Chapitre 3

Théorie des calculs pour les systèmes à temps continu

Ce chapitre correspond à une traduction (avec une légère adaptation) d'un chapitre d'un livre, signé en collaboration avec Manuel Campagnolo, qui paraîtra chez Springer Verlag “New Computational Paradigms”. Il s'agit d'un survol, issu de nombreuses discussions et d'échanges de points de vue avec Manuel, sur la théorie des calculs pour les systèmes à temps continu.

Nous présentons les théories des calculs des systèmes à temps continu. Ces théories permettent à la fois de comprendre la difficulté des questions relatives aux systèmes dynamiques à temps continu, et à la fois de comprendre la puissance des modèles analogiques à temps continu.

Nous résumons les résultats obtenus à ce jour, avec de nombreux pointeurs sur la littérature.

3.1 Introduction

Comme nous l'avons vu dans les chapitres 1 et 2, les systèmes dynamiques à temps continu apparaissent immédiatement dès que l'on tente de modéliser des systèmes qui évoluent sur un espace continu avec un temps continu. Ils peuvent même apparaître comme la description naturelle de systèmes à espace ou à temps discret. C'est une approche courante dans des domaines comme la biologie, la physique ou la chimie, où une grande population d'agents (comme des molécules ou des individus) est abstraite en des quantités réelles, qui peuvent être des proportions ou des données thermodynamiques [Hirsch et al., 2003], [Murray, 2002].

Il y a plusieurs approches qui ont mené à des théories pour les calculs à temps continu. Nous allons explorer en profondeur deux approches. La première, que nous nommerons *inspirée par les machines analogiques à temps continu*, possède ses racines dans les modèles de machines analogiques naturelles ou artificielles. La seconde, que nous nommerons *inspirée par les théories des systèmes à temps continu*, est d'un spectre plus large. Elle est issue de la recherche sur les systèmes à temps continu d'un point de vue de la calculabilité ou de la complexité. Les systèmes hybrides, les automates temporisés, ou les résultats de l'analyse récursive à propos de la calculabilité des solutions d'équations différentielles en sont par exemple des sources d'inspiration.

Un large domaine de problèmes reliés aux théories des calculs à temps continu est couvert par ces deux approches, dans des domaines comme la vérification (voir par exemple [Asarin et al., 1995]), la théorie du contrôle (voir par exemple [Branicky, 1995b]), la conception de circuits intégrés (voir par exemple [Mills et al., 2005]), les réseaux de neurones (voir par exemple [Orponen, 1997]) ou la théorie des calculs sur les réels (voir par exemple [Moore, 1996]).

À ses débuts, la théorie des calculs à temps continu se focalisait principalement sur les machines analogiques. Déterminer quels systèmes peuvent réellement êtres considérés comme des modèles de calculs est une question passionnante et intrigante, reliée la question philosophique de ce qu'est ou ce que n'est pas une machine programmable, ce qui en dehors du cadre de ce chapitre.

Néanmoins, une bonne part des toutes premières machines construites sont considérées comme des machines analogiques programmables. Cela inclut le célèbre Differential Analyser, construit pour la première fois sous la supervision de Vannevar Bush au MIT en 1931 [Bush, 1931], mais aussi le Finance Phalograph de Bill Phillips, et en dans un certain sens le planimètre d'Hermann en 1814, la Pascaline de Pascal en 1642, et même le mécanisme d'Anticythère de -87 avant JC : voir [Coward, 2006]. Les modèles de calculs à temps continu incluent aussi les réseaux de neurones formels, et plus généralement les systèmes qui peuvent êtres construits en utilisant de l'électronique analogique. Puisque les modèles continus apparaissent dès que l'on cherche à modéliser des grandes populations, nous pouvons spéculer qu'ils joueront un rôle prédominant comme modèles des systèmes massivement parallèles comme l'Internet [Papadimitriou, 2001] dans l'avenir : voir aussi le chapitre 2.

Le premier vrai modèle d'une machine à temps continu universelle est dû à Shannon [Shannon, 1941], qui l'a introduite comme un modèle du Differential Analyzer. Les années 50 et 60 ont donné lieu à la naissance d'une littérature importante sur la programmation de telles machines[1]. Il y a aussi une littérature relativement importante sur comment utiliser ces machines analogiques pour la résolution de problèmes discrets ou continus : voir par exemple [Vergis et al., 1986] et ses références. Cependant, la plupart de cette littérature est maintenant relativement peu pertinente, étant donnés les développements actuels de notre compréhension de la complexité et de la calculabilité.

La recherche sur les réseaux de neurones artificiels, bien qu'à ce jour essentiellement tournée vers les modèles à temps discret, a entraîné un changement de perspective, en raison des nombreuses relations établies avec les concepts modernes de la théorie de la calculabilité ou de la complexité [Orponen, 1997], [Orponen, 1994]. Une autre ligne de travaux a été motivée par les systèmes hybrides, en relation avec des questions sur la difficulté de leur vérification ou de leur théorie du contrôle : voir par exemple [Branicky, 1995b], [Asarin et al., 1995].

Les années récentes ont aussi vu un intérêt pour d'autres alternatives aux machines digitales classiques. Cela inclut les modèles à espace continu et à temps discret comme les réseaux de neurones artificiels [Orponen, 1997], les modèles optiques [Woods and Naughton, 2005], les machines basées sur des signaux [Durand-Lose, 2005] et le modèle de Blum Shub et Smale [Blum et al., 1998]. Plus généralement, cela inclue les modèles non classiques et plus ou moins réalistes ou futuristes comme les modèles d'automates cellulaires exotiques [Grigorieff and Margenstern, 2004], les calculs moléculaires ou naturels [Head, 1987], [Adleman, 1994], [Lipton, 1995], [Păun, 2002], les calculs par trous noirs [Hogarth, 1992], les calculs quantiques [Deutsch, 1985], [Gruska, 1997], [Shor, 1994], [Kieu, 2004].

Dans un sens, les modèles à temps discret sont relativement bien connus et compris grâce à la thèse de Church, qui affirme que tous les modèles raisonnables suffisamment puissant sont équivalents. Pour les systèmes à temps continu, la situation est loin d'être aussi claire, et il n'y a pas eu autant d'efforts pour tenter d'unifier les concepts, bien que certains résultats récents établissent l'équivalence entre des modèles apparemment distincts [Graça and Costa, 2003], [Graça, 2004], [Graça et al., 2005], et [Bournez et al., 2006], et laissent envisager qu'une théorie unifiée des calculs à temps continu est possible dans le futur.

Ce texte peut être considéré comme une mise à jour du survol de Orponen en 1997 [Orponen, 1997]. Comme l'écrit Orponen à la fin de son introduction, nous pouvons toujours dire que l'effet de l'imprécision ou du bruit sur les calculs analogiques est loin d'être compris, et qu'une théorie de la complexité robuste pour les modèles à temps continu a toujours à être réellement inventée. Cependant, nous le verrons, de nombreux progrès ont été obtenus dans ces directions dans la dernière décade.

Ce chapitre est organisé comme suit. Dans la section 3.2, nous rappelons les modèles à temps continu principaux. Dans les sections 3.3 et 3.4, nous discutons respectivement de la calculabilité et de la complexité des calculs à temps continu. Dans ces sections, nous nous focalisons principalement sur les systèmes à temps continu. Dans la section 3.5, nous nous intéressons aux effets des imprécisions et du bruit sur les calculs analogiques. Finalement, la section 3.6, est une conclusion avec des pistes et des directions pour des recherches futures dans le domaine des calculs à temps continu.

[1] Voir par exemple le très instructif *Analog Computer Museum* de Doug Coward sur Internet [Coward, 2006], et toute sa bibliographie. Observons que cette littérature révèle un art relativement oublié de la programmation des machines à temps continu, qui n'a rien à envier à la programmation actuelle

3.2 Modèles à temps continu

Avec à l'esprit une perspective historique, nous soulignons dans cette section plusieurs classes majeures de modèles à temps continu qui ont mené à des intérêts pour ce domaine de recherche. Ces modèles illustrent aussi des concepts comme les dynamiques continues, ou les entrées et les sorties.

3.2.1 Modèles inspirés par des machines analogiques

Le General Purpose Analog Computer de Shannon

La machine à temps continu probablement la plus connue est le *Differential Analyzer*, construit au MIT pour la première fois sous la supervision de Vannevar Bush [Bush, 1931] en 1931. L'idée d'assembler des périphériques réalisant des intégrations pour résoudre des équations différentielles peut être attribuée à Lord Kelvin en 1876 [Thomson, 1876]. Des machines mécaniques[2], et plus tard des Differential Analyzer électroniques ont été intensivement utilisés pour résoudre différentes équations différentielles provenant de problèmes d'ingénierie : voir par exemple [Bowles, 1996], ou plus généralement [Williams, 1996] pour des comptes-rendus historiques. A partir des années 1960, et l'invention du transistor, les analyseurs différentiels ont été progressivement écartés en faveur de la technologie digitale.

La première étude théorique des capacités de calcul des machines universelles à temps continu est due à Shannon. Shannon a proposé en 1941 [Shannon, 1941] le *General Purpose Analog Computer (GPAC)* comme un modèle théorique du Differential Analyzer de Vannevar Bush. Le modèle raffiné ensuite dans la série d'articles [Pour-El, 1974], [Lipshitz and Rubel, 1987], [Graça and Costa, 2003], [Graça, 2004], consiste simplement en des familles de circuits construits à partir des unités de base de la figure 3.1. En général, tous les types d'interconnexions ne sont pas autorisés, puisque cela conduirait à des comportements indésirables : par exemple, à des sorties non uniques. Pour plus de détails et de discussions, se référer à [Graça and Costa, 2003] et [Graça, 2002].

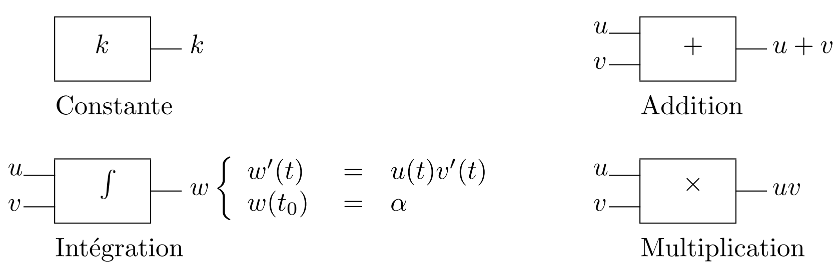


Fig. 3.1 – Les éléments de base d'un GPAC.

Shannon, dans son article original, mentionne déjà que les GPAC sait générer les polynômes, la fonction exponentielle, les fonctions trigonométriques, et leurs inverses. Plus généralement, Shannon prétend dans [Shannon, 1941] qu'une fonction est générée par un GPAC si et seulement si elle est différentiellement algébrique, c'est-à-dire si elle satisfait une équation différentielle algébrique de la forme

$$p\left(t, y, y', ..., y^{(n)}\right) = 0,$$

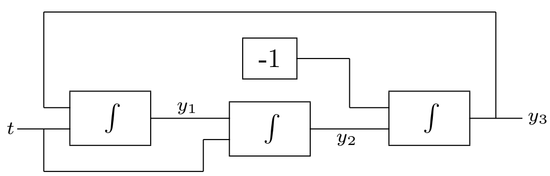
où p est un polynôme non nul en toutes ses variables. Par conséquent, si l'on note que la fonction Gamma $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ ou la fonction Zeta de Riemann $\zeta(x) = \sum_{k=0}^{\infty} \frac{1}{k^x}$ n'est pas différentiellement algébrique [Rubel, 1989], il en suit que la fonction Gamma et la fonction Zeta sont des exemples de fonctions qui ne peuvent pas êtres générées par un GPAC.

[2]Et aussi en MECANO : voir [Bowles, 1996].

Cependant, la preuve de Shannon qui relie les fonctions générées par un GPAC avec les fonctions différentiellement algébriques est incomplète (comme montré et partiellement corrigé par [Pour-El, 1974], [Lipshitz and Rubel, 1987]). Pour la classe de GPAC plus robuste considérée dans [Graça and Costa, 2003], la propriété plus forte suivante est vérifiée : une fonction scalaire $f : \mathbb{R} \to \mathbb{R}$ est générée par un GPAC si et seulement si elle est composante d'une solution d'un système $y' = p(t, y)$ où p est un vecteur de polynômes (elle est projection d'une solution d'un problème de Cauchy polynomial selon la terminologie du chapitre 1). Une fonction $f : \mathbb{R} \to \mathbb{R}^k$ est dite générée par un GPAC si et seulement si ses composantes le sont.

La fonction Γ est en fait bien calculable par un GPAC, si une notion de calcul inspirée par celle qui est présente en analyse récursive est considérée [Graça, 2004]. Les fonctions GPAC calculables dans ce sens correspondent précisément aux fonctions calculables sur les réels : voir [Bournez et al., 2006] ou le chapitre 4.

D'autres extensions du modèle original de GPAC de Shannon ont été considérées : en particulier Rubel a considéré dans [Rubel, 1993] le Extended Analog Computer (EAC), ou des opérations pour résoudre des problèmes aux bornes, ou pour prendre certaines limites infinies, ont été ajoutées. Nous renvoyons à [Mills, 1995] et à [Mills et al., 2005] pour la description d'implémentations électroniques du EAC de Rubel.



$$\begin{cases} y_1' = y_3 & \& \quad y_1(0) = 1 \\ y_2' = y_1 & \& \quad y_2(0) = 0 \\ y_3' = -y_1 & \& \quad y_3(0) = 0 \end{cases}$$

Fig. 3.2 – Génération de cos et sin par un GPAC : la version circuit est à gauche, et la version équation différentielle à droite. On a $y_1 = \cos$, $y_2 = \sin$, $y_3 = -\sin$.

Plus généralement, une discussion des circuits analogiques constitués de blocs élémentaires a été présentée récemment dans [Tucker and Zucker, 2007]. Cet article montre que la spécification équationnelle de tels circuits, ainsi que leur sémantique, peuvent être données en termes de points fixes d'opérateurs sur l'espace des flots continus. Avec de bonnes hypothèses, ces opérateurs sont contractants et une extension du théorème du point fixe de Banach pour les espaces métriques permet de garantir l'existence et l'unicité du point fixe. En outre, ce point fixe peut être prouvé continu et *concrètement* calculable lorsque les blocs élémentaires le sont.

Modèles de réseaux de Hopfield

Un autre modèle à temps continu bien connu est le modèle de réseaux de neurones artificiels proposé par John Hopfield en 1984 dans [Hopfield, 1984], qui peut s'implémenter avec des composants électroniques simples [Hopfield, 1984], ou par des périphériques optiques [Stoll and Lee, 1988].

Un réseau symétrique de Hopfield est constitué d'un nombre fini, disons n d'unités de calcul simple, ou *neurones*. L'architecture du réseau est donnée par un réseau (non orienté) dont les noeuds sont les neurones, et dont les arêtes sont étiquetées par des poids, les *poids synaptiques*. On peut supposer le graphe complet en remplaçant l'absence d'une connexion par une arête de poids nul.

L'état de chaque neurone i au temps t est donné par une valeur réelle $u_i(t)$. En partant d'un certain état initial $\mathbf{u}_0 \in \mathbb{R}^n$, la dynamique globale du réseau est définie par un système d'équations différentielles

$$C_i u_i'(t) = \sum_j W_{i,j} V_j - u_i/R_i + I_i,$$

où $V_i = \sigma(u_i)$, σ est une fonction saturante telle que $\sigma(u) = \alpha \tan u + \beta$, $W_{i,j} = W_{j,i}$ est le poids de l'arête entre i et j, et C_i, I_i, R_i sont des constantes [Hopfield, 1984].

Hopfield a montré dans [Hopfield, 1984], par un argument de fonction de Lyapunov, que de tels systèmes sont globalement asymptotiquement stables, i.e. de n'importe quel état initial, le système converge vers un état stable. En effet, si l'on considère par exemple la fonction énergie [Hopfield, 1984]

$$E = -\frac{1}{2}\sum_i\sum_j W_{i,j}V_iV_j + \sum_i \frac{1}{R_i}\int_0^{V_i} \sigma^{-1}(V)dV + \sum_i I_iV_i,$$

La fonction E est bornée et sa dérivée est négative. Par conséquent, l'évolution de l'état global du système est un déplacement dans l'espace des états à la recherche d'un minimum (qui peut être local) de E.

Ce comportement de convergence a été exploité par Hopfield pour de nombreuses applications, telles que des mémoires associatives, ou pour résoudre certains problèmes de décision comme le problème du voyageur de commerce [Hopfield, 1984], [Hopfield and Tank, 1985].

De tels réseaux sont en fait appelés réseaux de Hopfield en raison de leur analogie avec la version discrète du modèle introduite par le même auteur plus tôt. Cependant, la convergence de la dynamique continue peut aussi être vue comme une conséquence des résultats antérieurs de [Cohen and Grossberg, 1983] qui s'appliquent à des classes plus générales de systèmes dynamiques.

Une borne inférieure exponentielle sur le temps de convergence des réseaux de Hopfield à temps continu en fonction de leur dimension a été obtenue dans [Šíma and Orponen, 2003a]. De tels réseaux symétriques à temps continus peuvent simuler tout réseau de neurone récurrent à temps discret à états binaires, comme cela a été prouvé dans [Orponen and Šíma, 2000], [Šíma and Orponen, 2003b].

Réseaux de neurones à décharges

Selon [Maass, 1997b], si l'on classifie les modèles de réseaux de neurones artificiels en fonction de leur fonction d'activation et leur dynamique, on peut considérer qu'il y a trois générations. La première génération, avec des fonctions d'activation discontinues, inclue les perceptrons multicouches, les réseaux de Hopfield symétriques à temps discret, et les machines de Boltzmann (voir par exemple [Abdi, 1994] pour une introduction à tous ces modèles). Dans cette génération, les sorties sont digitales. La seconde génération n'utilise pas des fonctions à seuil pour calculer les sorties, mais des fonctions d'activation continues. Elle inclut les réseaux de neurones sigmoïdaux récurrents, ou les réseaux de fonctions à bases radiales, ou encore les réseaux de Hopfield à temps continu. Maintenant, les entrées et les sorties deviennent analogiques. La troisième génération, qui augmente encore le niveau de réalisme biologique [Maass and Bishop, 1998], est basée sur des neurones à décharges, et encode ses variables dans des différences temporelles entre pulsations. Cette troisième génération peut exhiber des dynamiques à temps continu.

Parmi les modèles mathématiques des neurones à décharges, les propriétés calculatoires du modèle qui suit ont été relativement bien étudiées. Un réseau de neurones à décharges est donné par un graphe orienté fini. À chaque noeud v (neurone) du graphe est associée une fonction à seuil $\theta_v : \mathbb{R}^+ \to \mathbb{R} \cup \{\infty\}$, et à chaque arête (u, v) (synapse) est associée une fonction de réponse $\epsilon_{u,v} : \mathbb{R}^+ \to \mathbb{R}$ et une fonction de poids $w_{u,v}$.

Pour un neurone v qui n'est pas un neurone d'entrée, on définit son ensemble F_v de temps de décharges récursivement. Le premier élément de F_v est $\inf\{t | P_v(t) \geq \theta_v(0)\}$, et pour tout $s \in F_v$, le prochain plus grand élément de F_v est $\inf\{t | t > s \text{ and } P_v(t) \geq \theta_v(t-s)\}$ où

$$P_v(t) = 0 + \sum_u \sum_{s \in F_u, s<t} w_{u,v}(s)\epsilon_{u,v}(t-s).$$

Le 0 ci-dessus, ici pour garantir que P_v est bien défini même si $F_u = \emptyset$ pour tout u avec $w_{u,v} \neq 0$, peut être remplacé par une fonction de biais. Des hypothèses inspirées par la biologie mènent à des restrictions sur les fonctions de réponses et de biais autorisées : voir [Maass, 1997b], [Maass, 1999], [Natschläger and Maass, 2002], ou [Maass, 2002], [Maass, 2003], pour des discussions sur le modèle.

L'étude de la puissance de calculs de tels réseaux a été initiée par [Maass, 1996a] selon plusieurs variantes. Des extensions bruitées du modèle ont été considérées dans [Maass, 1996b], [Maass, 1997a], ou encore dans l'article [Maass and Natschläger, 2000]. Un aperçu des résultats de complexité obtenus peut être trouvé dans le survol [Šíma and Orponen, 2003c]. Des restrictions motivées par l'implémentation matérielle des réseaux ont aussi été étudiées dans [Maass and Ruf, 1999].

Fonctions $\mathbb{R}$-récursives

Moore a proposé dans [Moore, 1996] une théorie des fonctions récursives, définie en analogie avec la théorie de la récursivité classique, et qui correspond à un ordinateur conceptuel analogique en temps continu. Comme nous le verrons, ce modèle à temps continu a en particulier la capacité de résoudre des équations différentielles, de façon similaire à un intégrateur analogique idéal dans un GPAC. Une discussion générale des discussions derrière la théorie des fonctions $\mathbb{R}$-récursives peut être trouvée dans [Mycka and Costa, 2005].

Une algèbre de fonctions

$$[B_1, B_2, ...; O_1, O_2, ...]$$

est le plus petit ensemble qui contient des fonctions de base $\{B_1, B_2, ...\}$ et qui est clos par certaines opérations $\{O_1, O_2, ...\}$, qui prennent une ou plusieurs fonctions dans la classe et en crée une nouvelle. Bien que des algèbres de fonctions aient été définies dans le contexte de la théorie de la récursivité sur les entiers, et aient été largement utilisées pour caractériser des classes de complexité ou de calculabilité [Clote, 1998], elles permettent aussi de définir des classes de fonctions à valeurs réelles.

Les fonctions $\mathbb{R}$-récursives ont été définies pour la première fois dans [Moore, 1996]. Elles sont données par l'algèbre de fonctions $\mathcal{M} = [0, 1, U; \mathsf{comp}, \mathsf{int}, \mathsf{minim}]$,[3] où U est l'ensemble des fonctions projection $U_i(\vec{x}) = x_i$, comp est la composition, int est une opération qui étant données les fonctions f et g retourne la solution du problème de Cauchy $h(\vec{x}, 0) = f(\vec{x})$ et $\partial_y h(\vec{x}, y) = g(\vec{x}, y, h)$ et minim retourne la plus petite racine $\mu_y f(\vec{x}, y)$ d'une fonction f donnée. Moore considère aussi l'algèbre plus faible $\mathcal{I} = [0, 1, -1, U; \mathsf{comp}, \mathsf{int}]$ et affirme son équivalence avec la classe des fonctions unaires générées par GPAC [Moore, 1996].

Beaucoup d'ensembles non-récursivement énumérables sont $\mathbb{R}$-récursifs. Puisque minim est l'opération dans $\mathcal{M}$ qui donne lieu à des fonctions non-calculables, une question naturelle est de savoir si minim peut se remplacer par une autre opération de l'analyse mathématique. Cela a été fait dans [Mycka and Costa, 2004] où minim est remplacé par l'opération lim, qui retourne la limite en l'infini des fonctions de l'algèbre. Ces auteurs stratifient $[0, 1, -1, U; \mathsf{comp}, \mathsf{int}, \mathsf{lim}]$ selon le nombre d'opérateur (η) utilisé dans les limites imbriquées, et relie la η-hiérarchie résultante avec les hiérarchies arithmétiques et analytiques. Dans [Loff et al., 2007a], il est prouvé que la η-hiérarchie ne s'effondre pas (voir aussi [Loff, 2007]), ce qui implique que les limites infinies et les intégrations du premier ordre ne sont pas des opérations interchangeables [Loff et al., 2007b].

L'algèbre $\mathcal{I}$ contient seulement des fonctions analytiques et n'est pas close par itération, comme prouvé dans [Campagnolo et al., 2000]. Cependant, si une extension θ réelle suffisemment lisse de la fonction de Heaviside est incluse parmi les fonctions de base de $\mathcal{I}$, alors $\mathcal{I} + \theta$ contient des extensions aux réels de toutes les fonctions primitives récursives.

La clôture des fragments de $\mathcal{I} + \theta = [0, 1, -1, \theta, U; \mathsf{comp}, \mathsf{int}]$ par des opérations comme les produits bornés, les sommes bornées et des récursions bornées, a été étudiée dans la thèse [Campagnolo, 2001] ainsi que dans les articles [Campagnolo et al., 2002], [Campagnolo, 2002], [Campagnolo, 2004].

En particulier, plusieurs auteurs ont étudié l'algèbre de fonctions $\mathcal{L} = [0, 1, -1, \pi, \theta, U;\ \mathsf{comp}, \mathsf{LI}]$ où l'opérateur LI est seulement capable de résoudre des équations différenielles *linéaires* (i.e., il restreint int au cas $\partial_y h(\vec{x}, y) = g(\vec{x}, y)\, h(\vec{x}, y)$). La classe $\mathcal{L}$ contient des extensions aux réels de toutes les fonctions élémentaires, [Campagnolo et al., 2002].

Plutôt que de se demander quelles fonctions sur $\mathbb{N}$ possèdent des extensions à $\mathbb{R}$ dans une algèbre de fonctions donnée, Bournez et Hainry considèrent les classes de fonctions sur $\mathbb{R}$ calculables en analyse récursive, et les caractérisent precisément en termes d'algèbres de fonctions. Cela a été fait pour les fonctions élémentairement calculables dans [Bournez and Hainry, 2005], qui sont caractérisées comme $\mathcal{L}$ clos par un schéma limite restreint. Cela a été étendu pour obtenir une caractérisation de la classe complète des fonctions calculables sur les réels [Bournez and Hainry, 2006], en ajoutant un schéma de minimisation restreint. Ces

[3] Nous considérons que l'opérateur int préserve l'analycité (voir [Campagnolo et al., 2000], [Campagnolo, 2002]).

résultats fournissent des caractérisations syntaxiques des fonctions calculables dans le cadre continu, ce qui est nettement plus naturel que les machines de Turing d'ordre supérieur de l'analyse récursive.

Une approche plus générale pour la complexité structurelle des classes réelles récursives, développée dans [Campagnolo and Ojakian, 2007], est basée sur la notion d'approximation. Cette notion a été utilisée pour étendre des des résultats de complexité de $\mathbb{N}$ vers $\mathbb{R}$, et en particulier pour caractériser $\mathcal{L}$.

D'une façon assez surprenante, tous ces résultats indiquent que des modèles distincts de calculs sur les réels (analyse récursive, et fonctions réelles récursives) peuvent se relier de façon élégante.

3.2.2 Modèles inspirés par les théories de systèmes à temps continu

Systèmes hybrides

Un nombre croissant de systèmes démontrent une certaine combinaison de comportements continus et discrets. L'investigation de ces systèmes a donné lieu à des nouveaux résultats en rapport avec la théorie des calculs à temps continu.

De nombreux modèles existent (voir par exemple la série de colloques *Hybrid Systems Computation and Control* ou la tentative de comparaison des modèles de [Branicky, 1995a]), mais on peut dire que les systèmes hybrides[4] sont essentiellement modélisés soit comme des équations différentielles avec des membres droits discontinus, ou par des équations différentielles avec des variables continues et discrètes, ou par des automates hybrides. Un automate hybride est un automate d'états finis étendu avec des variables. Son évolution est donnée par des transitions discrètes gardées entre les états (places) de l'automate, qui peuvent remettre à zéro certaines variables, et par la dynamique d'évolution de chacune des variables dans les places. Les propriétés typiques des systèmes hybrides qui sont considérées dans la littérature sont les propriétés d'atteignabilité, de stabilité ou de contrôlabilité.

En relation avec l'approche par des équations différentielles, Branicky a prouvé dans [Branicky, 1995b] que tout modèle de système hybride qui peut implémenter une horloge et les équations différentielles génériques peut simuler les machines de Turing. Asarin, Maler et Pnueli ont montré dans [Asarin et al., 1995] que les équations différentielles constantes par morceaux peuvent simuler les machines de Turing dans $\mathbb{R}^3$, alors que le problème de l'atteignabilité pour ces systèmes est décidable en dimension $d \leq 2$ [Asarin et al., 1995]. Les équations différentielles constantes par morceaux, comme beaucoup de modèles de systèmes hybrides, font montre de ce que l'on appelle le phénomène de Zénon : un nombre non-fini de transitions discrètes peut avoir lieu en temps fini. Cela a été utilisé dans [Asarin and Maler, 1998] pour prouver que les ensembles arithmétiques peuvent êtres reconnus en temps fini par ces systèmes. Leur puissance de calculs exacte a été caractérisée en termes de leurs dimensions dans [Bournez, 1999a] et dans [Bournez, 1999b]. Les arguments de [Asarin et al., 1995] basés sur le théorème de Jordan pour prouver la décidabilité des équations différentielles constantes par morceaux planaires ont été généralisés pour prouver la décidabilité des équations polynomiales planaires [Ceraens and Viksna, 1996] et des systèmes d'inclusions différentielles planaires [Asarin et al., 2001].

Il y a une littérature importante à propos de l'approche de modélisation par automates hybrides visant à déterminer la frontière entre décidabilité et non-décidabilité pour les propriétés d'atteignabilité, en fonction des types de dynamiques, des gardes et des remises à zéros autorisées. Les propriétés d'atteignabilité sont connues pour êtres décidables pour les automates temporisés [Alur and Dill, 1990]. Cela a été généralisé par la suite aux automates multitaux [Alur et al., 1995], à des classes spécifiques d'automates temporisés à mise à jour [Bouyer et al., 2000a], [Bouyer et al., 2000b], ou encore aux automates rectangulaires initialisés dans [Henzinger et al., 1998], [Puri and Varaiya, 1994], à chaque fois en utilisant un argument basé sur une bissimulation finie, similaire dans l'esprit à celui de [Alur and Dill, 1990]. Il y a une multitude de résultats d'indécidabilité, la plupart se basant sur la simulation de machines à deux compteur de Minsky. Par exemple, le problème de l'atteignabilité est semi-décidable mais non-décidable pour les automates hybrides linéaires [Alur et al., 1995], [Nicollin et al., 1993]. Le même problème est connu pour être indécidable pour

[4]"Hybride" est ici en référence au fait que les systèmes mélangent des évolutions discrètes et continues. Cela diffère de la littérature historique sur les calculs analogiques, où "hybride" fait souvent référence aux machines qui mélangent des composants analogiques et digitaux.

les automates rectangulaires avec au moins 5 horloges et une variable à deux taux [Henzinger et al., 1998], pour les automates temporisés avec deux horloges biaisées [Alur et al., 1995]. Pour des discussions, voir aussi [Asarin and Schneider, 2002]. Se référer à [Blondel and Tsitsiklis, 1999], [Collins and van Schuppen, 2004] ou à [Blondel and Tsitsiklis, 2000] pour d'autres propriétés que l'atteignabilité (par exemple la stabilité ou l'observabilité).

Les sytèmes hybrides *o-minimaux* sont des systèmes hybrides initialisés dont les ensembles et flots associés sont définissables dans une théorie o-minimale. Ces sytèmes possèdent toujours des bissimulations finies [Lafferriere and Pappas, 2000]. Cette notion et ce résultat peuvent s'étendrent à une classe plus générale de systèmes o-minimaux "non-déterministes" [Brihaye and Michaux, 2005], pour laquelle le problème de l'atteignabilité est indécidable dans le modèle de Turing, ainsi que dans le modèle de calculs de Blum Shub et Smale [Brihaye, 2006]. Des bornes supérieures ont été obtenues sur la taille des bissiumations finies pour les systèmes hybrides Pfaffiens [Korovina and Vorobjov, 2004] [Korovina and Vorobjov, 2006] en utilisant les techniques d'encodage par mots introduite dans [Brihaye and Michaux, 2005].

Théorie des automates

Il y a eu plusieurs tentatives d'étendre la théorie des automates classiques discrète au temps continu : cela est parfois appelé le programme général de Trakhtenbrot [Trakhtenbrot, 1995].

La première approche est reliée aux automates temporisés qui peuvent être vus comme des reconnaisseurs de langages [Alur and Dill, 1994]. De nombreux problèmes de décision spécifiques ont été considérés pour les automates temporisés : voir [Alur and Madhusudan, 2004], [Tripakis, 2003], [Finkel, 2005]. Les langages temporisés réguliers sont connus pour être clos par intersection, union, renommage mais non par complémentation. Le problème de l'appartenance, et le problème du vide sont décidables, alors que les problèmes de l'inclusion et de l'universalité sont indécidables. Leur clôture par mélange a été étudiée dans [Finkel, 2006]. Plusieurs variantes du théorème de Kleene ont été établies dans [Asarin et al., 1997], ou encore dans [Asarin, 1998], [Bouyer and Petit, 1999], [Asarin et al., 2002], [Bouyer and Petit, 2002], [Asarin and Dima, 2002]. Il y a eu des tentatives d'établir des lemmes de la pompe [Beauquier, 1998]. Un survol, avec discussions et problèmes ouverts reliés à cette approche peut être trouvé dans [Asarin, 2004].

Une théorie des automates indépendante et alternative a été développée par Trakhtenbrot et Rabinovich [Rabinovich and Trakhtenbrot, 1997], [Trakhtenbrot, 1999], [Rabinovich, 2003]. Ici les automates ne sont pas considérés comme des reconnaisseurs de langages, mais comme calculant des opérateurs sur des signaux. Un signal est une fonction des réels positifs ou nuls vers un alphabet fini (l'ensemble des états des canaux). La théorie des automates est étendue au temps continu, et il est argumenté que le comportement des périphériques finis est régi par les fonctions rétrospectives à mémoire finie. Elles sont prouvées insensibles à la vitesse, i.e. indépendantes par dilatation de l'axe du temps. Des propriétés de clôture des opérateurs sur les signaux sont établies, et la représentation des fonctions rétrospectives à mémoire finie par des diagrammes de transitions finis (transducteurs) est discutée. Voir aussi [Francisco, 2002] pour une présentation détaillée de la théorie de Trakhtenbrot et Rabinovich, et pour des discussions sur la représentation des fonctions rétrospectives à mémoire finie par des circuits.

Finalement, une autre approche indépendante est considérée dans l'article [Ruohonen, 2004], où une hiérarchie dans l'esprit de celle de Chomsky est établie pour des familles d'ensembles de fonctions continues par morceaux. Des équations différentielles, associées à des structures de mémoire spécifiques, sont utilisées pour reconnaître des ensembles de fonctions. Ruohonen montre que la hiérarchie résultante n'est pas triviale, et établi des propriétés de clôture et d'inclusions entre classes.

Autres modèles de calculs

En addition aux deux grandes approches précédentes, il y a un certain nombre d'autres modèles de calculs qui ont mené à des développements de la théorie des systèmes à temps continus.

La question de savoir si la théorie de la relativité générale d'Einstein admet des équations d'espace-temps qui permettent à un observateur de voir une éternité en un temps fini a été répondue par l'affirmative dans [Hogarth, 1992]. La question de savoir si cela implique qu'en principe des hyper-calculs peuvent être effectués a été investiguée dans [Earman and Norton, 1993], [Hogarth, 1996], [Hogarth, 1994], [Hogarth, 2006], [Etesi and Németi, 2002], [Németi and Andréka, 2006], [Németi and Dávid, 2006], et [Welch, 2006].

Certains modèles inspirés par des machines ne sont pas clairement discrets ou analogiques. Par exemple, la puissance des mécanismes planaires constitués de barres rigides contraintes à évoluer dans un plan et jointes à leurs extrémités par des rivets a attiré beaucoup d'attention en Angleterre et en France dans la fin du 19ième siècle, et dans les années 1940 en Russie. Un théorème attribué[5] à Kempke [Kempe, 1876] dit qu'ils sont capables de calculer toutes les fonctions algébriques : voir par exemple [Artobolevskii, 1964] ou [Svoboda, 1948].

3.3 Équations différentielles et propriétés

La plupart des modèles à temps continu décrits plus haut ont une dynamique continue décrite par des équations différentielles. Dans le GPAC de Shannon et dans les réseaux de Hopfield, l'entrée d'un calcul correspond à la condition initiale, alors que la sortie est respectivement, l'évolution temporelle ou l'état d'équilibre atteint du système. D'autres modèles sont des reconnaisseurs de langages. L'entrée est à nouveau une condition initiale, ou un contrôle initial, et la sortie est déterminée par une région acceptante dans l'espace des états du système.

Tous ces systèmes tombent par conséquent dans le cadre des systèmes dynamiques à temps continu. Plus généralement, tout modèle de calcul peut être vu comme un système dynamique.

Cette section vise à rappeler quelques résultats fondamentaux à propos des systèmes dynamiques et des équations différentielles, et nous discuterons comment les différents modèles peuvent êtres comparés dans ce cadre général.

3.3.1 Systèmes dynamiques et équations différentielles

Considérons que nous travaillons dans $\mathbb{R}^n$. Soit $f : E \to \mathbb{R}^n$, où $E \subset \mathbb{R}^n$ est ouvert. Une équation différentielle (ordinaire) est donnée par $y' = f(y)$ et une solution est une fonction différentiable $y : I \subset \mathbb{R} \to E$ qui satisfait l'équation.

Pour tout $x \in E$, le théorème fondamental d'existence et d'unicité (voir par exemple [Hirsch et al., 2003]) pour les équations différentielles affirme que si f est Lipschitz sur E, i.e. s'il existe K tel que $||f(y_1)-f(y_2)|| < k||y_1 - y_2||$ pour tout $y_1, y_2 \in E$, alors la solution du problème de Cauchy

$$y' = f(y), \qquad y(t_0) = x \tag{3.1}$$

existe et est unique sur un certain intervalle d'existence maximal $I \subset \mathbb{R}$. Dans la terminologie des systèmes dynamiques, $y(t)$ est nommée une trajectoire, $\mathbb{R}^n$ l'espace des phases, et la fonction $\phi(t, x)$, qui donne la position $y(t)$ de la solution au temps t avec la condition initiale x, est appelée le flot. Nous appelons orbite le graphe de y dans $\mathbb{R}^n$.

En particulier, si f est continûment différentiable sur E, alors les hypothèses du théorème d'existence unicité sont satisfaites [Hirsch et al., 2003]. La plupart de la théorie mathématique a été développée dans ce cas, mais elle peut s'étendre à conditions plus faibles. En particulier, si f est supposée seulement continue, alors l'unicité est perdue, mais l'existence est garantie : voir par exemple [Coddington and Levinson, 1972]. Si f est autorisée à être discontinue, alors la notion de solution nécessite d'être raffinée. L'approche la plus célèbre est celle de Fillipov [Filippov, 1988]. Certains modèles de systèmes hybrides utilisent des notions ad hoc (et distinctes) de solutions. Par exemple, une solution d'une équation différentielle constante par morceaux dans [Asarin et al., 1995] est une fonction continue dont la dérivée à droite satisfait l'équation.

De façon très générale, un système dynamique peut se définir comme l'action d'un sous-groupe $\mathcal{T}$ sur un espace X, i.e. par une fonction (un flot) $\phi : \mathcal{T} \times X \to X$ qui satisfait les deux équations suivantes

$$\phi(0, x) = x \tag{3.2}$$

[5]Le théorème est très souvent attribué à Kempke [Artobolevskii, 1964], [Svoboda, 1948], même si apparemment il n'a jamais prouvé cela exactement.

$$\phi(t, \phi(s, x)) = \phi(t + s, x). \tag{3.3}$$

Il est bien connu que les sous-groupes de $\mathcal{T}$ de $\mathbb{R}$ sont soit denses dans $\mathbb{R}$ ou isomorphes aux entiers. Dans le premier cas, le temps est dit continu, dans le second discret.

Puisque les flots obtenus par des problèmes de Cauchy de la forme (3.1) satisfont ces équations, ils correspondent à des systèmes dynamiques à espace et temps continus particuliers. Pas tous les systèmes dynamiques à espace et temps continus peuvent être mis sous la forme d'une équation différentielle, mais les problèmes de Cauchy de la forme (3.1) sont suffisamment généraux pour couvrir une très large classe de tels systèmes. En particulier, si ϕ est continûment différentiable, alors $y' = f(y)$, avec $f(y) = \frac{d}{dt}\phi(t, y)\big|_{t=0}$, décrit la dynamique du système.

Pour les systèmes à temps discret, nous pouvons supposer sans perte de généralités que $\mathcal{T}$ correspond aux entiers. L'analogue du problème de Cauchy (3.1) pour les systèmes à temps discret est une équation de récurrence du type

$$y_{t+1} = f(y_t), \qquad y_0 = x. \tag{3.4}$$

Un système dynamique dont l'espace est discret et qui évolue avec un temps discret est qualifié de digital, sinon d'analogique. Une classification de plusieurs modèles de calculs en fonction de la nature de leur espace et de leur temps peut être trouvée dans la figure 3.3.1.

Etats Temps	Discrets	Continus
Discret	Machines de Turing [Turing, 1936] Lambda calcul [Church, 1936] Fonctions récursives [Kleene, 1936] Systèmes de Post [Post, 1946] Automates cellulaires Automates à piles Automates d'états finis $\vdots$	Réseaux de neurones [Hopfield, 1984] à temps discret Réseaux de neurones de [Siegelmann and Sontag, 1994] Systèmes PCD de [Asarin et al., 1995] Machines de Blum Shub Smale [Blum et al., 1989] Machines optiques de [Woods and Naughton, 2005] Machines à signaux de [Durand-Lose, 2005] Reconnaisseurs dynamiques de [Moore, 1998a] $\vdots$
Continu	Modèle BDE [Dee and Ghil, 1984]	[Shannon, 1941] GPAC Réseaux de neurones de [Hopfield, 1984] à temps continu Systèmes hybrides de [Branicky, 1995b] Systèmes PCD de [Asarin et al., 1995] Automates temporisés [Alur and Dill, 1990] Fonctions $\mathbb{R}$-récursives [Moore, 1996] $\vdots$

FIG. 3.3 – Une classification de certains modèles de calculs, en fonction de leur espace et de leur temps.

3.3.2 Systèmes dissipatifs et non-dissipatifs

Un point x^* de l'espace des états est appelé un point d'équilibre si $f(x^*) = 0$. Si le système est en x^* il le restera. Il est dit stable si, pour tout voisinage U de x^*, il y a un voisinage W de x^* dans U tel que toute solution partant d'un point x de W est définie et est dans U pour tout temps $t > 0$. Le point est asymptotiquement stable si, en addition des propriétés ci-dessus, on a $\lim y(t) = x^*$ [Hirsch et al., 2003].

Des conditions locales sur la différentielle $Df(x^*)$ de f en x^* sont bien connues. Si en un point d'équilibre x^* toutes les valeurs propres de $Df(x^*)$ possèdent des parties réelles strictement négatives, alors x^* est asymptotiquement stable, et en outre les solutions dans le voisinage approchent x^* exponentiellement. Dans ce cas, x^* est nommé un point source. En outre, il est connu que dans un point stable d'équilibre x^*, aucune valeur propre de $Df(x^*)$ ne peut avoir une partie réelle strictement positive [Hirsch et al., 2003].

En pratique, le théorème de stabilité de Lyapunov s'applique de façon plus large (même si x^* n'est pas un point source). Il affirme que s'il existe une fonction continue V définie sur un voisinage de x^*, différentiable (sauf peut être en x^*) avec $V(x^*) = 0$, $V(x) > 0$ pour $x \neq x^*$, et $dV(x)/dt \leq 0$ pour $x \neq x^*$ alors x^* est stable. Si en outre on a aussi $dV(x)/dt < 0$ pour $x \neq x^*$, alors x^* est asymptotiquement stable : voir [Hirsch et al., 2003].

Si la fonction V satisfait ces conditions en tout point, alors le système est globalement asymptotiquement stable : quel que soit le point initial x, les trajectoires convergeront ultimement vers les minimaux locaux de V. Dans ce contexte, la fonction de Lyapunov V peut s'interpréter comme une énergie, et ses minimaux correspondent à des attracteurs du système dynamique. Ils constituent des sous-ensembles bornés de l'espace des phases vers lesquels toute région de conditions initiales de volume non-vide converge lorsque le temps croit.

Un système dynamique est dit dissipatif si le volume d'un ensemble dans l'espace des phases décroît sous l'action du flot. Les systèmes dissipatifs sont caractérisés par des attracteurs. Par opposition, un système dynamique est dit conservatif si ce volume est conservé. Par exemple, tous les systèmes hamiltoniens sont conservatifs en raison du théorème de Liouville [Arnold, 1989]. Les systèmes dynamiques conservatifs ne peuvent êtres globalement asymptotiquement stables [Arnold, 1989].

3.3.3 Calculabilité des solutions des équations différentielles

Nous revoyons maintenant certains résultats sur la calculabilité au sens de l'analyse récursive (voir par exemple [Weihrauch, 2000]) des solutions d'une équation différentielle.

En général, étant donnée une fonction calculable f, on peut chercher à comprendre si la solution du problème de Cauchy (3.1) est aussi calculable dans le sens de l'analyse récursive. Si on impose que la solution du problème de Cauchy est unique, alors cette solution est calculable. Formellement, si f est calculable sur $[0,1] \times [-1,1]$ et $y' = f(t,y)$, $y(0) = 0$ possède une solution unique sur $[0,b]$, $0 < b \leq 1$, alors la solution y est calculable sur $[0,b]$ [Pour-El and Richards, 1979].

Cela est vrai pour les problèmes de Cauchy n-dimensionnels, si l'unicité des solutions est toujours parmi les hypothèses [Ruohonen, 1996].

Cependant, la calculabilité des solutions est perdue dès que l'unicité des solutions est relâchée, même en dimension 1. En effet, [Pour-El and Richards, 1979] montrent qu'il existe une fonction calculable f (même calculable en temps polynomial) $f : [0,1] \times [-1,1] \to \mathbb{R}$, telle que l'équation $y' = f(t,y)$, avec $y(0) = 0$, possède des solutions non-uniques, mais aucune des solutions n'est calculable sur tout intervalle de temps fermé.

Des phénomènes similaires ont lieu pour d'autres équations naturelles : l'équation d'onde en dimension 3 (qui est une équation différentielle partielle), avec des données initiales calculables, peut avoir une solution unique qui est calculable nulle part[6] [Pour-El and Richards, 1981], [Pour-El and Zhong, 1997]. Observons que, même si f est supposée calculable et analytique, et la solution est unique, il peut se produire que l'intervalle maximal (α, β) d'existence de la solution soit non-calculable [Graça et al., 2006]. La même question est ouverte pour f polynomiale. Ces auteurs montrent, cependant que si f et f' sont calculables, alors la solution du problème de Cauchy $y' = f(y,t)$, $y(0) = x$, pour x calculable, est calculable sur son intervalle maximal d'existence. Se référer à [Pour-El and Richards, 1989], [Ko, 1983] pour plus de résultats d'indécidabilité, et aussi à [Ko, 1983], [Ko, 1991] pour des considérations de complexité, et pas seulement de calculabilité.

3.3.4 Indécidabilité statique

Comme cela est observé dans [Asarin, 1995] ou dans [Ruohonen, 1997b], il est facile, mais souvent pas très informatif, d'obtenir des résultats d'indécidabilité avec des systèmes dynamiques à temps continu.

[6] Cependant, dans tous ces cas, les problèmes étudiés sont mal posés : soit la solution est non unique, ou elle est instable. L'ajout de conditions naturelles visant à régulariser et éviter que le problème soit mal posé mène à la calculabilité [Weihrauch and Zhong, 2002].

À titre d'illustration, rappelons l'exemple suivant tiré de [Ruohonen, 1997b], qui discute le problème de la détection d'évènements (étant donnée une équation différentielle $y' = f(t, y)$, avec une valeur initiale $y(0)$, décider si une certaine condition $g_j(t, y(t), y'(t)) = 0, j = 1, \cdots, k$ se produit à un temps t dans un intervalle donné I), fixons n'importe quelle machine de Turing universelle $\mathcal{M}$. La suite $f_0, f_1, \cdots$ de rationnels définie par

$$f_n = \begin{cases} 2^{-m} & \text{si} \quad \mathcal{M} \text{ s'arrête en } m \text{ étapes sur l'entrée } n \\ 0 & \text{si} \quad \mathcal{M} \text{ ne s'arrête pas sur l'entrée } n \end{cases}$$

n'est pas une suite calculable de rationnels, mais est une suite calculable de réels, selon la nomenclature de [Pour-El and Richards, 1989]. Maintenant, la détection de l'évènement $y(t) = 0$ pour l'équation différentielle $y' = 0$, étant donnés n, et la valeur initiale $y(0) = f_n$, est indécidable parce que $f_n = 0$ est indécidable.

Une autre modification peut être obtenue comme suit. Considérons la fonction lisse

$$g(x) = f_{\lfloor x+1/2 \rfloor} e^{-\tan^2 \pi x},$$

qui est calculable sur $[0, \infty)$. La détection de l'évenement $y_1(t) = 0$ pour le problème de Cauchy

$$\begin{cases} y_1' & = & g(y_2) - 1 \\ y_2' & = & 0 \end{cases}$$

avec $y_1(0) = 1$, $y_2(0) = n$, où n est un entier positif ou nul est alors indécidable sur $[0, 1]$.

Beaucoup des résultats d'indécidabilité fournis par l'analyse récursive sont d'une certaine façon dans cet esprit [Asarin, 1995].

3.3.5 Indécidabilité dynamique

Pour pouvoir discuter de façon plus profonde la calculabilité par des équations différentielles, nous nous focalisons maintenant sur des équations différentielles qui encodent les transitions d'une machine de Turing plutôt que le résultat de tout un calcul directement[7]. Typiquement, on part d'une fonction calculable (simple) injective qui code chaque configuration d'une machine de Turing M par un point de $\mathbb{R}^n$. Soit x le codage de la configuration initiale de M. On recherche alors une fonction $f : E \subset \mathbb{R}^{n+1} \to \mathbb{R}^n$ telle que la solution de $y'(t) = f(y, t)$, avec $y(0) = x$, au temps $T \in N$, donne le codage de la configuration de M après T étapes. Nous verrons, dans le reste de cette section, que f peut être restreinte à être de basse dimension, a être lisse ou analytique, ou a être définie sur un domaine compact.

Plutôt que de formuler la propriété plus haut comme une simulation d'une machine de Turing, on peut le formuler en termes de résultat d'atteignabilité. Étant donné un problème de Cauchy donné par f et x, et une région $A \subset \mathbb{R}^n$, on s'intéresse à décider s'il existe un $t \geq 0$ tel que $y(t) \in A$, i.e., si le flot partant de x atteint A. Il est clair que si f simule une machine de Turing dans le sens précédent, alors le problème de l'atteignabilité pour cette classe de systèmes est indécidable (considérer A comme le codage des configurations d'arrêt de M). Par conséquent, l'atteignabilité est une autre façon d'étudier la calculabilité des équations différentielles, et un résultat négatif est souvent la conséquence d'un résultat de simulation de machines de Turing. De façon similaire, l'indécidabilité de la détection d'évènements est une conséquence de résultats de simulation de machines de Turing.

La calculabilité des ensembles invariants ou atteignables a été étudiée par [Collins, 2005] pour les systèmes dynamiques généraux à temps continu, et dans [Collins and Lygeros, 2005] pour les systèmes hybrides.

En général, interpréter les machines de Turing comme des systèmes dynamiques procure une signification physique à la machine de Turing qui n'est pas donnée par la vision classique de von Neumann [Campagnolo, 2001]. Cela montre aussi que nombre de caractéristiques qualitatives des systèmes dynamiques (analogiques ou non-analogiques), par exemple les questions relatives aux bassins d'attractions, aux comportements chaotiques ou même périodiques, ne sont pas calculables [Moore, 1990]. Réciproquement, cela amène dans le cadre des machines de Turing et de la calculabilité en général de nombreuses questions traditionnellement associées aux systèmes dynamiques. Cela inclut par exemple les relations entre l'universalité et le chaos

[7] Cela est nommé indécidabilité dynamique par [Ruohonen, 1993].

[Asarin, 1995],des conditions nécessaires pour l'universalité [Delvenne et al., 2004], la compréhension de la frontière du chaos [Legenstein and Maass, 2007], la calculabilité de l'entropie [Koiran, 2001], et les relations avec la propriété de shadowing [Hoyrup, 2006]. On observera que les machines de Turing sont présentées comme des systèmes dynamiques déjà dans [Minsky, 1967].

3.3.6 Plongement de machines de Turing en temps continu

Le plongement des machines de Turing dans les systèmes dynamiques à temps continu est souvent réalisé en deux étapes. Les machines de Turing sont tout d'abord plongées dans des systèmes dynamiques à espace continu et à temps discret, et les systèmes obtenus sont ensuite plongés dans des systèmes à temps et espace continus.

La première étape peut être réalisée en basses dimensions avec des dynamiques simples : les articles [Moore, 1990], [Ruohonen, 1993], [Branicky, 1995b], et [Ruohonen, 1997b] considèrent des systèmes dynamiques généraux, l'article [Koiran et al., 1994] considère des fonctions affines définies par morceaux, l'article [Siegelmann and Sontag, 1995] des réseaux de neurones sigmoïdaux linéaires seuillés, [Koiran and Moore, 1999] des fonctions analytiques en forme close undimensionnelles, qui peuvent être robustes [Graça et al., 2005], et l'article [Kurganskyy and Potapov, 2005] des fonctions unidimensionnelles restreintes définies par morceaux.

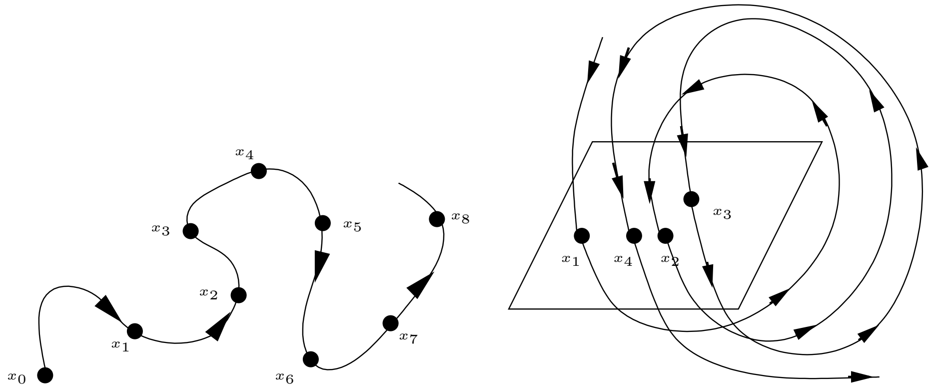


Fig. 3.4 – Discrétisation par stroboscope (à gauche) et par une section de Poincaré (à droite) d'un système dynamique à temps continu.

Pour la deuxième étape, l'astuce usuelle est de construire un système dynamique à temps et espace continu dont une discrétisation correspond au système dynamique à espace continu qui réalise le plongement.

Il existe plusieurs façons classiques de discrétiser un système à temps et espace continu : voir la figure 3.3.6. Une façon est d'utiliser un stroboscope virtuel : le flot $x_t = \phi(t, x)$, lorsque t est restreint aux entiers, définit les trajectoires d'un système dynamique à temps discret. Une autre possibilité est via l'utilisation d'une section de Poincaré : la suite x_t des intersections des trajectoires avec (par exemple) une hypersurface peut fournir le flot d'un système dynamique à temps discret : voir [Hirsch et al., 2003].

L'opération inverse, appelée suspension, est généralement réalisée en étendant et en rendant lisses les équations, et requiert généralement le recours à des équations en dimension supérieure.

Cela explique pourquoi les machines de Turing sont simulées par des systèmes dynamiques à temps continu lisses de dimension 3 dans [Moore, 1990], [Moore, 1991], [Branicky, 1995b] ou par des équations différentielles constantes par morceaux de dimension 3 dans [Asarin et al., 1995], alors que l'on sait les simuler par des fonctions affines par morceaux de dimension seulement 2 en temps discret [Koiran et al., 1994]. Il est connu que les équations différentielles constantes par morceaux de dimension 2 ne peuvent pas[8] simuler les machines de Turing arbitraires [Asarin et al., 1995], alors que la question de savoir si les fonctions affines par morceaux de dimension 1 peuvent simuler les machines de Turing arbitraires est ouverte.

[8]Voir aussi les généralisations de ce résultat dans [Ceraens and Viksna, 1996] et [Asarin et al., 2001].

D'autres simulations de machines de Turing par des dynamiques à temps continu incluent la simulation robuste avec des équations différentielles polynomiales de [Graça et al., 2005] et [Graça et al., 2007]. Ce résultat constitue une version améliorée de la simulation des machines de Turing par des fonctions réelles récursives de [Campagnolo et al., 2000], où il est montré que des classes de fonctions lisses mais non-analytiques sont closes par itérations. Observons que, bien que la solution d'une équation différentielle polynomiale soit calculable sur son domaine maximal d'existence (voir la section 3.3.3), les résultats de simulation montrent que le problème de l'atteignabilité est indécidable pour les équations différentielles polynomiales.

En addition des machines de Turing, d'autres modèles discrets peuvent être simulés par des équations différentielles. La simulation de machines à 2 compteurs peut se réaliser en dimension 2 et même en dimension 1 au prix d'équations différentielles discontinues [Ruohonen, 1997b]. La simulation d'automates cellulaires peut être réalisée par des équations différentielles partielles à l'aide de fonctions C^∞ [Omohundro, 1984].

Observons que les calculs réversibles des machines de Turing (ou des machines à compteurs, ou à registres) peuvent être simulés par des équations différentielles ordinaires avec des solutions uniques en arrière (backward-unique) [Ruohonen, 1993].

Les systèmes dynamiques à temps continu peuvent à leur tour être plongés dans d'autres systèmes à temps continu. Par exemple, [Maass et al., 2007] prouve qu'une large classe S_n de systèmes d'équations différentielles sont universelles pour les calculs analogiques sur les entrées qui dépendent du temps dans le sens suivant : un système de cette classe peut répondre à une entrée externe $u(t)$ par la dynamique de toute équation différentielle ordinaire d'ordre n de la forme $z^{(n)}(t) = G(z(t), z'(t), \cdots, z^{(n-1)}(t)) + u(t)$, si une fonction de retour sans mémoire et une fonction de lecture convenables sont ajoutées. Comme les équations différentielles ordinaires d'ordre n peuvent simuler les machines de Turing, les systèmes de S_n ont la puissance d'une machine de Turing universelle. Mais puisque G est arbitraire, les systèmes de S_n peuvent en fait simuler toute réponse continue et dynamique à une fonction d'entrée. En outre, ce résultat est valide pour le cas où les entrées et les sorties sont supposées bornées.

3.3.7 Une discussion

Une technique clé pour plonger l'évolution temporelle d'une machine de Turing dans un flot est d'utiliser des "horloges continues" comme dans [9] [Branicky, 1995b].

L'idée est de partir d'une fonction $f : \mathbb{R} \to \mathbb{R}$, qui préserve les entiers, et de construire l'équation différentielle sur $\mathbb{R}^3$

$$\begin{aligned} y_1' &= c(f(r(y_2)) - y_1)^3 \theta(\sin(2\pi y_3)) \\ y_2' &= c(r(y_1) - y_2)^3 \theta(-\sin(2\pi y_3)) \\ y_3' &= 1. \end{aligned}$$

Ici $r(x)$ est l'analogue d'une fonction entière : $r(x) = n$ pour $x \in [n - 1/4, n + 1/4]$ pour un entier n, et $\theta(x)$ vaut 0 pour $x \leq 0$, $\exp(-1/x)$ pour $x > 0$, et c est une constante bien choisie.

La variable y_3 sert uniquement à fournir la variable temps t. Supposons $y_1(0) = y_2(0) = x$. Pour $t \in [0, 1/2]$, $y_2' = 0$, et donc y_2 est gardé fixé à x. Maintenant, si $f(x) = x$, il est clair que y_1 sera gardé fixé à x. Si $f(x) \neq x$, alors $y_1(t)$ approchera $f(x)$ sur cet intervalle de temps, et en se référant aux calculs dans [Campagnolo, 2001], si c est choisi suffisamment grand, on peut être sûr que $|y_1(1/2) - f(x)| \leq 1/4$. Par conséquent, nous aurons $r(y_1(1/2)) = f(x)$. Maintenant, pour $t \in [1/2, 1]$, les rôles sont inversés $y_1' = 0$, et donc y_1 est gardé fixé à la valeur de $f(x)$. La variable y_2 approche $f(x)$, et nous aurons $r(y_2(1)) = f(x)$. L'équation a un comportement similaire pour tous les intervalles qui suivent de la forme $[n, n + 1/2]$, $[n{+}1/2, n{+}1]$, et donc à tout temps entier t, $f^{[t]}(x) = r(y_1(t))$. [10] [Loff et al., 2007a] propose une construction similaire qui retourne $f^{[\lfloor t \rfloor]}(x)$ pour tout $t \in \mathbb{R}$.

En d'autres termes, la construction plus haut transforme une fonction sur $\mathbb{R}$ en une équation différentielle (en dimension supérieure) qui simule ses itérations. Cependant, pour faire cela, nous avons eu besoin de $\theta(\sin(2\pi y_3))$ qui sert d'horloge. À la base, le système dynamique à temps continu obtenu a une "senteur

[9] Branicky attribue l'idée d'un calcul en deux phases à [Brockett, 1989] et [Brockett, 1991]. Une astuce similaire est présente dans [Ruohonen, 1993]. Nous ne suivons pas en fait l'article original [Branicky, 1995b] mais sa présentation dans [Campagnolo, 2001].

[10] $f^{[t]}(x)$ dénote la $t$ème itération de f sur x.

hybride". Même si le flot est lisse (i.e. est C^∞) par rapport au temps, l'orbite n'admet pas de tangente en tout point puisque y_1 et y_2 sont alternativement constants.

On peut chercher à dépasser cette restriction en se limitant aux simulations de machines de Turing par des flots et des fonctions analytiques. Bien qu'il soit connu que les fonctions analytiques sur des domaines non bornés peuvent simuler la fonction de transition de n'importe quelle machine de Turing [Koiran and Moore, 1999], ce n'est que très récemment qu'il a été montré que les machines de Turing peuvent êtres simulées par des flots analytiques (et même polynomiaux) sur des domaines non bornés [Graça et al., 2005]. Il serait désirable d'étendre ce type de résultats aux domaines compacts. Cependant, il est conjecturé dans [Moore, 1998b] que cela n'est pas possible, i.e. qu'aucune fonction analytique sur un domaine fini dimensionnel compact peut simuler une machine de Turing arbitraire via un codage des entrées et des sorties raisonnable.

3.3.8 Contractions du temps et de l'espace

Les machines de Turing sont simulées par les équations différentielles en temps réel : par exemple, dans les constructions évoquées plus haut, l'état $y(T)$ au temps $T \in N$ encode la configuration de la machine de Turing après T étapes. Cependant, puisque les systèmes à temps continu peuvent exhiber des phénomènes de contractions d'espace et de temps, des machines de Turing accélérantes[11] [Davies, 2001], [Copeland, 1998], [Copeland, 2002] et même des machines de Turing avec oracles peuvent êtres simulées en des temps arbitrairement courts.

En effet, en suivant la présentation de [Ruohonen, 1993], dénotons un système à temps continu par un triplet (F, n, A), où F défini l'équation différentielle $y' = F(y)$ sur $\mathbb{R}^n$, avec l'ensemble acceptant A : une entrée x est acceptée si et seulement si la trajectoire partant avec la condition initiale x atteint A.

Une telle machine peut être accélérée : la substitution $t = e^u - 1$ par exemple change la machine $\mathcal{M} = (F, n, A)$ en $((G, 1), n + 1, A \times \mathbb{R})$ où

$$\frac{dg}{du} = G(g(u), u) = F(g(u))e^u \text{ and } g(u) = y(e^u - 1),$$

ce qui donne une accélération du temps exponentielle. Observons que les dérivées de la solution par rapport à la nouvelle variable temporelle u sont exponentiellement plus grandes.

La substitution $t = \tan(\pi u/2)$ mène à une accélération temporelle infinie, i.e. compresse tout calcul, même infini, en un calcul sur l'intervalle de temps fini $0 \leq u < 1$. Maintenant, les dérivées vont vers l'infini lors du calcul.

Pour obtenir une contraction de l'espace, remplaçons l'état $y(t)$ de la machine $\mathcal{M} = (F, n, A)$ par $r(t) = y(t)e^{-t}$. Cela donne une machine dont l'espace est exponentiellement réduit $((H, 1), m + 1, H_1)$ où

$$\frac{dr}{dt} = H(r(t), t) = F(r(t)e^t)e^{-t} - r(t)$$

et

$$H_1 = \{(e^{-t}q, t) | q \in A \text{ and } t \geq 0\}.$$

Bien entendu, cette transformation réduit exponentiellement la distance entre les trajectoires, ce qui impose une précision supplémentaire pour les distinguer.

Des résultats de dureté pour les niveaux des hiérarchies arithmétiques et analytiques pour plusieurs problèmes de décisions pour les systèmes à temps continu sont déduits de constructions similaires dans [Ruohonen, 1993], [Ruohonen, 1994], [Moore, 1996], ou dans [Asarin and Maler, 1998]. Des résultats de complétude, tout comme une caractérisation exacte du pouvoir de reconnaissance des équations différentielles constantes par morceaux en fonction de leur dimension sont obtenus dans [Bournez, 1999a], [Bournez, 1999b]. Observons que de tels phénomènes sont des instances du phénomène dit de Zénon dans la littérature des systèmes hybrides [Alur and Dill, 1990] [Asarin and Maler, 1998].

[11] Des possibilités similaires de simulation des machines de Turing accélérantes en mécanique quantique sont discutées dans [Calude and Pavlov, 2002].

Il peut être remarqué que les constructions précédentes fournissent des résultats d'indécidabilité seulement pour des fonctions sur des domaines infinis ou semi-ouverts, puisque les entiers positifs ou nuls, qui correspondent aux temps entiers des machines de Turing sont envoyés sur les intervalles de la forme $[0, 1)$. Une construction par l'analyse a été développée dans [Ruohonen, 1997a] pour montrer qu'une compression temporelle est possible sur un domaine compact de la forme $[0, 1]$, avec une fonction G continue, et bornée sur $[0, 1]$ mais non-différentiable. Il en suit que le problème de la détection d'évènements est indécidable même pour les fonctions continues sur un domaine compact [Ruohonen, 1997a].

L'indécidabilité est écartée, cependant, si la fonction G est suffisamment lisse sur $[0, 1]$ (disons $\mathcal{C}^1$), si à la fois G et les conditions initiales sont calculables, et si une condition d'acception suffisamment robuste est considérée. En effet, des problèmes comme la détection d'évènements deviennent décidables, puisque le système peut être simulé effectivement [Ruohonen, 1997a].

Plutôt que de plonger des machines de Turing dans des systèmes dynamiques continus, il est naturel de se demander s'il y aurait une meilleure notion de calcul et de complexité pour les systèmes dynamiques provenant de modèles de notre monde physique. Nous nous penchons sur cette question dans la section qui suit.

3.4 Vers une théorie de la complexité

Nous discutons maintenant un certain nombre de vues sur la complexité des systèmes dynamiques continus. Nous considérons des systèmes généraux et la question de la difficulté de simuler des systèmes à temps continu par un modèle digital. Nous nous focalisons alors sur les systèmes dissipatifs, où les trajectoires convergent vers des attracteurs. En particulier, nous discutons l'idée que le temps de calcul devrait être le temps naturel de l'équation différentielle. Finalement, nous présentons des résultats de complexité pour des systèmes à temps continu plus généraux qui correspondent à des classes de fonctions réelles récursives.

3.4.1 Systèmes généraux

Dans [Vergis et al., 1986], les auteurs se demandent si les machines analogiques peuvent être plus efficaces que les machines digitales. Vergis et al. postulent la validité de la thèse de Church forte pour les modèles analogiques, qui instanciée sur ces modèles, clame que le temps requis par un ordinateur digital pour simuler un machine analogique est borné par une fonction polynomiale des ressources utilisées par la machine analogique. Vergis et al. prétendent que la thèse de Church forte est prouvable pour les systèmes dynamiques à temps continu définis par une équation différentielle $y' = f(y)$ Lipchitzienne.

Les ressources utilisées par un ordinateur analogique incluent l'intervalle temporel d'opération, disons $[0, T]$, la taille du système, qui peut être mesurée par $\max_{t\in[0,T]} ||y(t)||$, tout comme une borne sur les dérivées de y. Par exemple, le poids, le temps d'opération, et les forces appliquées sont toutes des ressources utilisées par une particule décrite en mécanique Newtonienne [Vergis et al., 1986].

L'affirmation plus haute nécessite de préciser ce que l'on entend par "simulation". Dans [Vergis et al., 1986], il est considéré que le problème de Cauchy $y' = f(y)$, $y(0) = x$ est simulé si, étant donné une précision ε, on peut calculer une approximation de $y(T)$ avec une erreur au plus ε. En utilisant la méthode d'Euler, et en supposant que les erreurs d'arrondis sont majorées par σ, la borne totale sur l'erreur est donnée par

$$||y(T) - y_N^*|| \leq \frac{h}{\lambda}\left[\frac{R}{2} + \frac{\sigma}{h^2}\right](e^T\lambda - 1), \tag{3.5}$$

où y_N^* est l'approximation après N étapes, h est le pas utilisé, λ la constante de Lipchitz pour f sur $[0, T]$, et $R = \max\{||y''(t)||, t \in [0, T]\}$. De cette borne (3.5), Vergis *et al.* concluent que le nombre N d'étapes nécessaires pour simuler le système par la méthode d'Euler est polynomial en R et $\frac{1}{\varepsilon}$. Ils utilisent ce fait pour affirmer que la thèse de Church forte est valide pour les équations différentielles Lipchitziennes.

Cependant, N est exponentiel en T, et T correspond au temps d'opération de la machine analogique modélisée. Cela rend les arguments de [Vergis et al., 1986] pas totalement convaincants, comme cela est souligné par exemple explicitement dans [Orponen, 1997].

Plus récemment, Smith a discuté dans [Smith, 2006] si des phénomènes d'hypercalculs sont possibles avec le problème des n corps en mécanique. En particulier, il a montré que cette dépendance exponentielle en T peut être éliminée. Comme le fait observé Smith, toutes les méthodes classiques numériques d'ordre fixe pour résoudre les équations différentielles souffrent du même problème de dépendance exponentielle en T. Cependant, en considérant une combinaison de méthodes de Runge-Kutta avec des ordres qui varient linéairement en T, il est effectivement possible de construire une méthode qui requiert un nombre d'étapes N polynomial en T, dès que la valeur absolue de chaque composante de f, y et la valeur absolue de chacune des dérivées partielles de f par rapport à chacun de ses arguments, ayant un degré total de différentiation k, est en $(kT)^{\mathcal{O}(k)}$ [Smith, 2006]. Les implications pour la thèse de Church forte sont discutées dans [Smith, 2006] et dans [Bournez, 2006] (voir aussi l'annexe A).

La même question peut être étudiée dans le cadre de l'analyse récursive. Lorsque $f : [0,1] \times [-1,1] \to \mathbb{R}$ est calculable en temps polynomial, et satisfait une forme faible de la condition de Lipchitz, l'(unique) solution y sur $[0,1]$ du problème de Cauchy $y' = f(t,y)$, $y(0) = 0$ est toujours calculable en espace polynomial [Ko, 1983]. Cependant, résoudre une équation différentielle en temps polynomial avec cette condition faible de Lipchitz est essentiellement aussi difficile que de résoudre un problème PSPACE-complet, puisqu'il existe une fonction f calculable en temps polynomial, qui satisfait cette condition, pour laquelle la solution y n'est pas calculable en temps polynomial sauf si $P = PSPACE$ [Ko, 1983], [Ko, 1991].

Les résultats de Ko ne sont pas directement comparables aux bornes polynomiales de Smith. En analyse récursive, l'entrée est le nombre de bits de précision. Si la borne sur l'erreur d'approximation de $y(t)$ est mesurée en bits, i.e. si $\varepsilon = 2^{-d}$, alors le nombre d'étapes N requis dans [Smith, 2006] est exponentiel en d.

Si f est analytique, alors la solution de $y' = f(y)$ est aussi analytique. Dans ce cas, les méthodes par pas peuvent êtres évitées. C'est l'approche suivie dans [Müller and Moiske, 1993], où il est prouvé que si f est analytique et calculable en temps polynomial alors la solution est calculable en temps polynomial.

En résumé, bien que la thèse de Church forte soit vérifiée pour les équations différentielles analytiques dans ce sens, elle n'a pas été complètement établie pour les systèmes généraux d'équations différentielles Lipchitziennes. La possibilité de calculs hyper-polynomiaux par des équations différentielles ne peut pas être écartée, au moins dans le principe à ce jour. Pour des discussions informelles sur la thèse de Church en relation avec les systèmes analogiques, voir aussi [Aaronson, 2005] et [MacLennan, 2001].

Plusieurs auteurs ont montré que certains problèmes de décision ou d'optimisation (par exemple la connectivité de graphes, la programmation linéaire) peuvent être résolus par des systèmes dynamiques spécifiques. Des exemples et des références peuvent êtres trouvés dans [Vergis et al., 1986], [Brockett, 1991], [Faybusovich, 1991a], [Helmke and Moore, 1994], et [Ben-Hur et al., 2002].

3.4.2 Systèmes dissipatifs

Nous nous focalisons maintenant sur les systèmes dissipatifs, en présentant deux approches. La première concerne essentiellement les réseaux de neurones formels, comme les réseaux de Hopfield. Elle se base sur les notions de la complexité de circuits, et mène à des bornes inférieures sur la complexité de tels réseaux. La seconde concerne les phénomènes de convergence vers les attracteurs, et considère les fonctions d'énergie comme des façons de mesurer la complexité des processus à temps continu.

Lorsque l'on considère des systèmes dissipatifs, comme les réseaux de neurones de Hopfield, l'approche suivante pour définir une théorie de la complexité devient naturelle. On considère des familles $(C_n)_{n \in N}$ de systèmes à temps continu, chaque C_n pour une certaine longueur d'entrée $n \geq 0$. Étant donnée une entrée (digitale) $w \in \{0,1\}^*$, on considère l'évolution du système C_n sur (un certain codage de) l'entrée w, où n est la longueur de w. Elle va finir ultimement par converger vers un état stable, qui sera considéré comme le résultat du calcul.

Cette notion de calcul, inspirée par les circuits, est la plus courante dans la littérature qui concerne la théorie de la complexité des réseaux de neurones : voir par exemple le survol [Šíma and Orponen, 2003c]. Dans ce cadre, les réseaux symétriques à temps continu de Hopfield avec une fonction d'activation linéaire seuillée peuvent simuler les réseaux de neurones récurrents binaires à temps discret arbitraires, avec un surcoût au plus linéaire [Orponen and Šíma, 2000], [Šíma and Orponen, 2003b].

On peut penser que cela va contre l'intuition, puisque de tels réseaux symétriques, qui sont contraints par une fonction d'énergie de Lyapunov, peuvent seulement exhiber des phénomènes de convergence, et donc ne peuvent même pas réaliser un simple bit alternant. Cependant, la convergence des systèmes à temps continu peut être exponentiellement longue en la taille du système [Šíma and Orponen, 2003a], et donc la simulation est réalisée en utilisant un sous-réseau qui fournit 2^n pulsations d'horloge avant de converger.

Les langages reconnus par des familles de réseaux de Hopfield à temps discret ont été prouvés dans [Orponen, 1996] comme correspondant à la classe de complexité (non uniforme) $PSPACE/poly$ pour des poids d'interconnexions arbitraires, et $P/poly$ pour des poids de tailles polynomialement bornées. Par conséquent, les familles de réseaux de Hopfield symétriques à temps continu ont au moins cette puissance. Ces bornes inférieures peuvent ne pas être optimales, puisque des bornes supérieures pour les dynamiques à temps continu ne sont pas connues [Šíma and Orponen, 2003b], [Šíma and Orponen, 2003c].

Intéressons nous maintenant aux systèmes dissipatifs avec une fonction de Lyapunov E.

Gori et Meer [Gori and Meer, 2002] considèrent un modèle de calcul qui a la capacité de trouver les minimisateurs (i.e. les points de minimum local ou global) de E. Pour éviter que la complexité d'un problème soit cachée dans la description de E, cette fonction doit être facile à calculer. Dans ce cadre, un problème Π est considéré comme facile s'il existe une fonction E unimodale (i. e. tous les minimisateurs de E sont des minimisateurs globaux) telle que la solution de Π puisse être obtenue facilement à partir du minimum global de E.

Plus précisément, Gori et Meer font l'investigation dans [Gori and Meer, 2002] d'un modèle où un problème Π sur les réels est considéré comme résolu s'il existe une famille $(E_n)_n : \mathbb{R}^n \times \mathbb{R}^{q(n)} \to \mathbb{R}$ de fonctions d'énergie, donnée par une famille uniforme de programmes en droite ligne (straight line programe) (q est un polynôme fixe), et une autre famille $(N_n)_n$ de programmes en droite ligne, telles que, pour toute entrée d, une solution $\Pi(d)$ du problème peut être calculée en utilisant $N_{q(n)}(w^*)$, à partir d'un minimisateur global w^* de $w \to E_n(d, w)$.

Ils introduisent deux classes U et NU, en analogie avec P et NP en complexité classique. U correspond au cas ci-dessus où, pour tout d, $w \to E_n(d, w)$ est unimodal, par opposition à NU, le cas général.

Des notions de réductions sont introduites, et il est montré qu'un problème d'optimisation naturel (trouver le minimum d'une fonction objective linéaire sur un ensemble défini par des contraintes polynomiales multivariées quadratiques) est NU-difficile. Il est montré qu'il existe des problèmes (artificiels) NU complets. Ces idées sont généralisées pour obtenir une hiérarchie polynomiale, avec des problèmes complets [Gori and Meer, 2002].

Le cadre proposé est en fait relativement abstrait, en évitant plusieurs problèmes reliés avec ce que certains pourraient attendre d'une vraie théorie de la complexité pour les calculs en temps continu. Néanmoins, il offre le grand avantage de ne pas se baser sur aucune mesure de complexité spécifique à propos des trajectoires. Voir la discussion très intéressante dans [Gori and Meer, 2002].

Cependant, il semble nécessaire de comprendre la complexité d'approcher les minima des fonctions d'énergie, qui correspondent aux équilibres des systèmes dynamiques. Des premières étapes vers ces directions ont été proposées dans l'article [Ben-Hur et al., 2002], où l'on se restreint aux systèmes dissipatifs avec convergences exponentielles. Rappelons que si x^* est une source, alors la vitesse de convergence vers x^* est du type

$$|x(t) - x^*| \equiv e^{-\lambda t}$$

où $-\lambda$ est la plus grande partie réelle d'une valeur propre de $Df(x^*)$. Cela signifie que $\tau = 1/\lambda$ est un temps caractéristique naturel de l'attracteur : tout les $\tau \log 2$ unités de temps, un nouveau bit de l'attracteur est calculé.

Pour les systèmes considérés dans [Ben-Hur et al., 2002], chaque source possède une région d'attraction, dans laquelle les trajectoires deviennent prisonnières. On définit le temps de calcul comme $t_c = max(t_c(\epsilon), t_c(U))$, où $t_c(\epsilon)$ est le temps requis pour atteindre un ϵ voisinage d'un attracteur, et $t_C(U)$ le temps requis pour atteindre sa région d'attraction. Alors $T = \frac{t_c}{\tau}$ est une mesure de complexité sans dimension, invariante par toute contraction temporelle linéaire.

Deux algorithmes à temps continu, MAX pour calculer le maximum de n nombres, et $FLOW$ pour résoudre le problème du flot maximal ont été étudiés dans ce cadre dans [Ben-Hur et al., 2002]. MAX a été montré comme appartenant à la classe de complexité $CLOG$ (temps logarithmique continu) et $FLOW$ à CP (temps polynomial continu). Les auteurs conjecturent que CP correspond aux temps polynomial classique [Ben-Hur et al., 2002]. Les deux problèmes MAX et $FLOW$ sont des instances spécifiques d'un flot proposé dans [Faybusovich, 1991b] pour résoudre les problèmes de programmation linéaire, étudié plus profondément dans [Ben-Hur et al., 2003] et [Ben-Hur et al., 2004a]. Des variations sur les définitions des classes de complexité, tout comme des moyens d'introduire des classes de complexité non-déterministes en relations avec les attracteurs chaotiques ont aussi été discutés dans l'article [Siegelmann and Fishman, 1998].

3.4.3 Complexité et fonctions réelles récursives

Les fonctions réelles récursives constituent un moyen commode d'analyser la puissance calculatoire de certaines opérations sur les fonctions réelles. En outre, étant donné un modèle à temps continu, si l'on sait prouver son équivalence avec une algèbre de fonctions, alors des propriétés du modèles peuvent être établies par induction sur l'algèbre de fonctions.

Puisque de nombreuses classes de complexité en espace et en temps ont des caractérisations récursives sur $\mathbb{N}$ [Clote, 1998], les résultats de complexité structurelle à propos des opérations discrètes peuvent impliquer des bornes inférieures et supérieures sur la complexité des fonctions réelles récursives. Cette approche a été suivie dans [Campagnolo et al., 2002] pour montrer que $\mathcal{L}$ contient des extensions des fonctions élémentaires, et a été poussée plus avant dans [Campagnolo, 2004] pour obtenir des classes plus faibles qui se relient à la hiérarchie en espace exponentiel. Cela permet de mieux comprendre la complexité de certains systèmes dynamiques. Par exemple, $\mathcal{L}$ correspond à des cascades de profondeur finie, chaque niveau dépendant de ses propres variables et des sorties des niveaux précédents.

Plusieurs résultats visant à transférer des questions de complexité sur les entiers vers des questions sur les réels ont été discutés auparavant. Concernant la théorie de la complexité, la question $P = NP$ en complexité classique a été investiguée en utilisant des fonctions réelles récursives par Costa et Mycka. En particulier, ils proposent deux classes de fonctions réelles récursives telles que leur séparation impliquerait $P \neq NP$ dans [Costa and Mycka, 2006], [Mycka and Costa, 2006]. Plus généralement, une partie du programme de Costa et Mycka, explicitement exprimé dans [Mycka and Costa, 2005] et [Mycka and Costa, 2007], consiste à utiliser la théorie de la récursion sur les réels pour établir des liens entre la complexité et la calculabilité et l'analyse en mathématiques.

3.5 Robustesse aux bruits et aux imprécisions

Jusqu'à ce point, nous avons considéré des calculs en temps continu dans des espaces idéalisés sans aucun bruit. La plupart des résultats que nous avons discutés ont été établis en ignorant l'impact du bruit ou des imprécisions dans les systèmes à temps continu. Cela constitue une faiblesse critiquée de façon récurrente dans le domaine, déjà observée par Orponen dans son survol [Orponen, 1997]. Observons que considérer des calculs en précision non-bornée est fortement analogue à considérer des machines de Turing à rubans infinis : la construction de machines à rubans infinis n'est pas possible, comme la construction de machines à précision infinie.

Bien qu'ils n'y aient pas eu de réelles percées vis-à-vis de ces questions en ce qui concerne spécifiquement les systèmes à temps continu, des développements intéressants concernant les effets du bruit et de l'imprécision ont été réalisés à propos des systèmes analogiques à temps discret. Aussi, dans cette section, nous élargissons notre propos pour discuter aussi de résultats relatifs à des systèmes à temps discret. Nous pensons que ces résultats peuvent être généralisés, ou tout du moins donnent des pistes pour être généralisés, aux systèmes à temps continu, bien que cela reste souvent à faire.

Nous nous focalisons dans un premier temps sur les systèmes à espace borné, à propos desquels une conjecture de tout le monde (folklore conjecture) clame que la robustesse implique la décidabilité. Nous présentons des résultats qui appuient cette conjecture, et des résultats qui semblent aller dans le sens opposé. À la fin de cette section, nous discutons les systèmes à temps continu avec un espace non-borné.

Les techniques communes pour simuler une machine de Turing par un système dynamique sur un espace borné nécessitent le codage des configurations de la machine de Turing en des nombres réels. Ces simulations sont détruites si les nombres réels, ou les fonctions impliquées ne sont pas représentées avec une précision infinie. Cela a mené à conjecture de tout le monde, populaire en particulier dans la communauté de la vérification, que l'indécidabilité n'a pas lieu pour les systèmes "réalistes", "non-précis", "bruités", "flous", "robustes". Voir par exemple [Fränzle, 1999], ou [Foy, 2004] pour différentes discussions dans le sens de cette conjecture, et [Asarin, 2006] pour des discussions des arguments formels ou informels qui mènent ou ont mené à cette conjecture.

Il n'y a pas de consensus à propos de ce qu'est un modèle de bruit réaliste. Une discussion de ce sujet nécessite de faire recours à la question de quels sont les bons modèles de notre monde physique. En l'absence d'un modèle de bruit universellement accepté, on peut cependant considérer différents modèles pour le bruit, l'imprécision, ou des conditions de régularité, et étudier les propriétés des systèmes résultants.

En particulier, il y a eu plusieurs tentatives de prouver que les systèmes analogiques bruités sont au mieux équivalents aux automates finis. Brockett montre dans [Brockett, 1989] que les systèmes dynamiques à temps continu peuvent simuler les automates finis arbitraires. En utilisant des arguments topologiques basés sur des relations d'équivalence par homotopies et les transformations de Deck associées, il montre réciproquement dans [Brockett, 1994] qu'à certains systèmes à temps continu dissipatifs on peut associer un automate.

Dans [Maass and Orponen, 1998], Maass et Orponen prouvent que la présence d'un bruit probabiliste réduit la puissance d'une large classe de modèles analogiques à temps discret sur un domaine compact aux langages réguliers. Cela étend un résultat dans [Casey, 1996], [Casey, 1998] pour le cas particulier où la sortie est supposée parfaitement fiable (i.e. $\rho = 1/2$ dans ce qui suit).

L'idée de Maass et d'Orponen est de remplacer une dynamique en temps discret parfaite de type $x_{i+1} = f(x_i, a_i)$, où a_i est le symbole d'entrée au temps i, sur un domaine compact, par une dynamique probabiliste

$$Probability(x_{i+1} \in B) = \int_{q \in B} z(f(x_i, a_i), q) d\mu,$$

où B est n'importe quel Borélien. Ici z est un noyau de densité qui reflète un bruit arbitraire, qui est supposé équicontinu par morceaux : pour tout ϵ, il existe δ tel que pour tout r, p, q, $\|p - q\| \leq \delta$ implique $|z(r,p) - z(r,q)| \leq \epsilon$.

Dénotons par $\pi_x(q)$ la distribution des états après que la chaîne x ait été lue, partant d'un état initial q. Une condition d'acception robuste est considérée : un langage L est reconnu, s'il existe $\rho > 0$ tel que $x \in L$ si et seulement si $\int_F \pi_{xu}(q) d\mu \geq 1/2 + \rho$ pour un certain $u \in \{U\}^*$, et $x \notin L$ si et seulement si $\int_F \pi_{xu}(q) d\mu \leq 1/2 - \rho$ pour tout $u \in \{U\}^*$, où U est un symbole de blanc, et F l'ensemble des états acceptants.

L'observation principale est alors que l'espace des fonctions $\pi_x(.)$ peut être partitionné en un nombre fini de classes C telles que deux fonctions $\pi_x(.)$ et $\pi_y(.)$ dans la même classe satisfont $\int_r |\pi_x(r) - \pi_y(r)| d\mu \leq \rho$, de telle sorte que deux mots x, y correspondant à la même classe satisfont $xw \in L$ si et seulement si $yw \in L$ pour tous mots w.

En fait, pour n'importe quel bruit usuel et commun, comme un bruit gaussien, qui n'est pas nul sur une partie suffisamment large de l'espace des états, de tels systèmes ne sont même pas capables de reconnaître tous les langages réguliers [Maass and Sontag, 1999]. Ils reconnaissent précisément les langages définis de [Rabin, 1963], comme cela est démontré dans [Maass and Sontag, 1999], [Ben-Hur et al., 2004b]. Si le support du bruit est borné, alors tous les langages réguliers peuvent être reconnus [Maass and Orponen, 1998]. Les circuits à temps continu avec feedbacks dans [Maass et al., 2007] ont la même puissance de calcul quand ils sont sujet à un bruit borné.

De façon alternative à l'approche de bruit probabiliste de Maass et Orponen, le bruit peut se modéliser par l'introduction d'un bruit non-déterministe.

À un système dynamique discret parfait (déterministe) S défini par $x_{i+1} = f(x_i)$, associons le système S_ϵ non-déterministe ϵ-perturbé dont les trajectoires sont les suites $(x_n)_n$ avec $\|x_{i+1} - f(x_i)\| \leq \epsilon$. Écrivons $Reach[S](x, y)$ (respectivement $Reach_n[S](x, y)$) s'il existe une trajectoire de S allant de x vers y (resp. en $i \leq n$ étapes).

La vérification algorithmique des propriétés de sûreté est reliée très étroitement au problème du calcul des états atteignables. Étant donné S, avec un sous-ensemble d'états initiaux S_0, écrivons $Reach[S]$ pour l'ensemble des y tels que $Reach[S](x,y)$ pour un certain $x \in S_0$. Étant donné une propriété sur les états p (c'est-à-dire une propriété p qui est soit vraie soit fausse en chaque point s de l'espace), soit $[[\neg p]]$ l'ensemble des états s en lesquels p est fausse. Alors S est sûr (c'est-à-dire p est un invariant) si et seulement si $Reach[S] \cap [[\neg p]] = \emptyset$ (voir par exemple [Alur et al., 1995] et [Nicollin et al., 1993]).

Si la classe de systèmes considérés est telle que la relation $Reach_n[S](x,y)$ est récursive[12], alors $Reach[S]$ est récursivement énumérable, puisque $Reach[S] = \bigcup_n Reach_n[S]$. Plusieurs articles ont visé à prouver que $Reach[S]$ est en fait récursif pour les classes de systèmes dynamiques robustes, pour différentes notions de robustesse. Nous revoyons maintenant ces articles.

Fränzle observe dans [Fränzle, 1999] que le calcul de $Reach[S_\epsilon]$ par la formule $Reach[S_\epsilon] = \bigcup_n Reach_n[S_\epsilon]$ termine toujours, si $Reach[S_\epsilon]$ possède un diamètre fortement fini : il existe un nombre infini de points dans $Reach[S_\epsilon]$ à distance mutuelle au moins ϵ. Cela est exclu par exemple sur un domaine borné. Il en suit que si nous appelons robuste un système qui est soit non sûr, ou tel qu'une ϵ-perturbation est sûre pour un certain ϵ, alors la sûreté est décidable pour les systèmes robustes sur un domaine compact [Fränzle, 1999].

Considérons, comme dans [Puri, 1998], la relation définie par $Reach_\omega[S] = \bigcap_{\epsilon>0} Reach[S_\epsilon]$, qui correspond aux états qui restent atteignables lorsque le bruit converge vers 0. Asarin et Bouajjani montrent que pour une large classe de systèmes dynamiques à temps discret et à temps continu (les machines de Turing, les fonctions affines par morceaux, les équations différentielles constantes par morceaux), $Reach_\omega[S]$ est co-récursivement énumérable. En outre, toute relation co-récursivement énumérable est de la forme $Reach_\omega[S]$ pour un certain système S de chacune de ces classes [Asarin and Bouajjani, 2001]. Il suit de la première observation que si nous appelons robuste un système tel que $Reach[S] = Reach_\omega[S]$, alors calculer $Reach[S]$ est décidable pour les systèmes robustes [Asarin and Bouajjani, 2001].

Asarin et Collins considèrent dans [Asarin and Collins, 2005] un modèle de machine de Turing exposée à un faible bruit stochastique, dont la puissance de calcul est caractérisée comme Π_2^0. Il est intéressant de comparer ce résultat avec les résultats précédents ou un bruit faible non-déterministe mène seulement à Π_1^0, c'est-à-dire aux ensembles co-récursivement énumérables.

Nous nous tournons maintenant vers des résultats qui vont dans le sens contraire de la conjecture que robustesse implique décidabilité.

Un premier exemple est que la sûreté des systèmes reste indécidable si la relation de transition des systèmes considérés est ouverte, comme cela est prouvé dans [Henzinger and Raskin, 1999] [Asarin, 2006]. Cependant, la question pour un bruit uniforme non-déterministe borné inférieurement est une question ouverte [Asarin, 2006].

Le bruit peut aussi être introduit en perturbant les trajectoires. Gupta Henzinger et Jagadeesan proposent dans [Gupta et al., 1997] de considérer une métrique sur les trajectoires des automates temporisés, et de supposer que si un système accepte une trajectoire, il doit aussi accepter les trajectoires du voisinage. Ils montrent que cette notion de robustesse n'est pas suffisante pour éviter l'indécidabilité de la complémentation des automates temporisés. Henzinger et Raskin prouvent dans [Henzinger and Raskin, 1999] que les principaux résultats d'indécidabilité pour la vérification des systèmes hybrides restent indécidables pour les systèmes robustes dans ce sens.

Finalement, nous revoyons un résultat récent pour les systèmes dynamiques à temps continu avec un espace d'états non-borné. Graça, Campagnolo et Buescu ont prouvé récemment que les équations différentielles polynomiales peuvent simuler de façon robuste les machines de Turing en temps réel. Plus précisément, considérons que $\theta : \mathbb{N}^3 \to \mathbb{N}^3$ soit la fonction de transition d'une machine de Turing M dont les configurations sont encodées dans $\mathbb{N}^3$. Alors, il existe un $\epsilon > 0$, une solution f d'une équation différentielle polynomiale, et une condition initiale $f(0)$ telle que la solution de $y' = f(t,y)$ encode l'état de M après t étapes avec une erreur d'au plus ϵ. En outre, cela reste vrai sur le voisinage de tout entier t, même si f et si la condition initiale $f(0)$ sont perturbés. Bien entendu, ce type de simulation requiert un espace d'états non-borné.

[12] Récursive en x, y et en n.

3.6 Conclusion

Par ce survol du domaine de la théorie des calculs pour les systèmes à temps continu, nous voyons qu'elle fournit des éclairages pour des domaines divers comme la vérification, la théorie du contrôle, la conception de circuits intégrés, les machines analogiques, la théorie de la récursion, la théorie des équations différentielles, et la complexité.

Nous avons tenté de fournir une présentation synthétique et systématique de tous les modèles principaux et des résultats sur les systèmes à temps continu. Dans la dernière décade, de nombreux nouveaux résultats ont été établis, ce qui prouve que c'est un domaine de recherche actif. Nous avons présenté des développements récents de la théorie des calculs à temps continu vis-à-vis de la calculabilité, la complexité, la robustesse au bruit et aux imprécisions, et nous avons identifié plusieurs problèmes ouverts. Pour conclure, nous voudrions discuter de plusieurs directions pour des recherches futures.

Calculabilité.

Il n'est pas clair si un concept unificateur comme la thèse de Church-Turing existe pour les systèmes à temps continu. Bien qu'il ait été montré que certains modèles à temps continu possèdent des capacités hypercalculatoires, tous ces résultats se basent sur l'utilisation d'une certaine quantité infinie de ressources, comme le temps, l'espace, la précision ou l'énergie. En général, il est souvent conjecturé que les modèles à temps continu "raisonnables" ne peuvent pas calculer plus que les machines de Turing. Cela mène à la question de savoir si les calculs à temps continu par des systèmes physiquement réalistes peuvent êtres aussi puissants que les modèles digitaux. Nous avons vu que si l'on se restreint aux systèmes à temps continu qui évoluent sur un espace d'états borné, et qui sont sujets à du bruit, alors ils deviennent comparables aux automates finis. Puisque les systèmes à temps continu robustes et analytiques peuvent simuler les machines de Turing avec un espace non-borné, nous pensons que les calculs digitaux et analogiques sont également puissants du point de vue de la calculabilité. En outre, comme nous l'avons vu, plusieurs résultats récents établissent l'équivalence entre les fonctions calculables par des équations différentielles polynomiales, les fonctions calculables par GPAC, et les fonctions calculables dans le sens de l'analyse récursive. Ce type de résultats renforce l'idée qu'il pourrait y avoir un cadre unifié pour les calculs à temps continu, similaire à celui qui existe en théorie de la calculabilité classique.

Nous pensons qu'un paradigme général de calcul à temps continu idéal ne devrait faire appel qu'à des fonctions analytiques, puisque ces dernières sont souvent considérées comme les plus acceptables d'un point de vue physique. Les systèmes dynamiques continus sont des modèles naturels pour représenter de nombreux phénomènes à temps continu. Les systèmes continus classiques, comme l'équation de van der Pol, les systèmes de Lotka Volterra, les équations de Lorenz, ou tous les exemples des chapitres 1 et 2 sont décrits en utilisant des équations différentielles analytiques, et même à membre droit polynomial. Ces arguments reliés à la physique, combinés avec les propriétés de calcul des systèmes d'équations différentielles polynomiales nous mènent à suggérer que le modèles à temps continu des équations différentielles définies par des problèmes de Cauchy polynomiaux est un candidat possible pour un paradigme général des calculs à temps continu. Cette idée mérite plus d'attention.

Complexité

Nous avons vu plusieurs pistes pour construire une théorie de la complexité pour les systèmes à temps continu. À vrai dire, à ce jour, le travail est largement en cours, et il n'y a pas d'accord général entre les auteurs sur les définitions de base comme le temps de calculs, ou la taille des entrées. Les résultats établis dans la section 3.4 sont soit dérivés de concepts intrinsèques aux systèmes à temps continu considérés, ou reliés à la théorie classique de la complexité. Puisque l'analyse récursive est un cadre bien établi et bien compris pour l'étude des problèmes de complexité pour les systèmes continus, nous pensons que mieux comprendre les relations qui existent entre les différentes approches et l'analyse récursive du point de vue de la complexité est de première importance. Il y a de nombreux problèmes qui restent ouverts à propos de bornes supérieures sur la puissance des systèmes à temps continu. Par exemple, des bornes supérieures ne sont pas connues pour les réseaux de neurones de Hopfield, ou pour les systèmes d'équations différentielles Lipchitziennes, ce qui compromet la validité de la thèse de Church forte. Nous avons vu que cette thèse peut peut-être être prouvée pour les équations différentielles analytiques. Cela mène à se demander si une théorie des calculs à temps continu basée sur les équations différentielles définies par des problèmes de Cauchy polynomiaux pourrait s'étendre en une théorie de la complexité.

L'analyse récursive permet aussi d'étudier la complexité des fonctions réelles. Un des domaines de recherche les plus intriguant consiste à comprendre les liens qui existent entre les fonctions réelles récursives et la complexité classique, pour établir par exemple des traductions des problèmes ouverts de la complexité dans le cadre de l'analyse.

Robustesse

Nous avons vu que très peu de recherche a été fait à ce jour en ce qui concerne les effets de bruits ou d'imprécision sur les calculs à temps continu. À ce jour, l'essentiel des résultats concerne les systèmes à temps discret, et leur transfert au cas du temps continu mérite de l'attention. On peut aussi se demander comment la puissance des systèmes analogiques croit avec leur précision. La question a été formulée et formalisée pour les systèmes à temps discret, en particulier pour les reconnaisseurs dynamiques dans [Moore, 1998b], mais la plupart de la recherche dans cette direction reste à faire. De nombreuses questions ouvertes surgissent si l'on demande si les résultats d'indécidabilité pour les systèmes à temps continu restent vrais pour les systèmes robustes. Cela est de première importance dans le domaine de la vérification par exemple, puisque cette question est reliée de façon très proche à la question de la preuve de la terminaison des procédures automatiques de vérification. Une meilleure compréhension des hypothèses avec lesquelles du bruit mène à de la décidabilité ou de l'indécidabilité est nécessaire. Par exemple, un bruit non-déterministe pour les systèmes ouverts n'écarte pas l'indécidabilité, mais la question est ouverte pour un bruit uniforme non-déterministe borné inférieurement [Asarin, 2006].

Nouveaux modèles

D'une façon plus spéculative, comme les systèmes à temps continu arrivent naturellement lorsque l'on cherche à modéliser les grandes populations (voir le chapitre 2), une question fascinante est de comprendre si la théorie des calculs en temps continu peut réellement aider à comprendre les modèles massivement parallèles comme l'Internet. Nous pensons que la théorie des calculs pour les systèmes à temps continu peut être au coeur de la prochaine génération de la complexité et de la calculabilité.

Bibliographie

[Aaronson, 2005] Aaronson, S. (2005). NP-complete problems and physical reality. *ACM SIGACT News*, 36(1) :30–52.

[Abdi, 1994] Abdi, H. (1994). A neural network primer. *Journal of Biological Systems*, 2 :247–281.

[Adleman, 1994] Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. *Science*, 266 :1021–1024.

[Alur et al., 1995] Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T. A., Ho, P. H., Nicollin, X., Olivero, A., Sifakis, J., and Yovine, S. (1995). The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1) :3–34.

[Alur and Dill, 1990] Alur, R. and Dill, D. L. (1990). Automata for modeling real-time systems. In *Automata, Languages and Programming, 17th International Colloquium*, volume 443 of *Lecture Notes in Computer Science*, pages 322–335. Springer-Verlag.

[Alur and Dill, 1994] Alur, R. and Dill, D. L. (1994). A theory of timed automata. *Theoretical Computer Science*, 126(2) :183–235. Fundamental Study.

[Alur and Madhusudan, 2004] Alur, R. and Madhusudan, P. (2004). Decision problems for timed automata : A survey. In Bernardo, M. and Corradini, F., editors, *Formal Methods for the Design of Real-Time Systems, International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM-RT 2004, Bertinoro, Italy, September 13-18, 2004, Revised Lectures*, volume 3185 of *Lecture Notes in Computer Science*, pages 1–24. Springer.

[Arnold, 1989] Arnold, V. I. (1989). *Mathematical methods of classical mechanics*, volume 60 of *Graduate texts in Mathematics*. Springer, second edition.

[Artobolevskii, 1964] Artobolevskii, I. (1964). *Mechanisms for the generation of plane curves*. Macmillan, New York. Translated by R.D. Wills and W. Johnson.

[Asarin, 2004] Asarin (2004). Challenges in timed languages : From applied theory to basic theory. *Bulletin of the European Association for Theoretical Computer Science*, 83 :106–120.

[Asarin and Collins, 2005] Asarin and Collins (2005). Noisy Turing machines. In *ICALP : Annual International Colloquium on Automata, Languages and Programming*.

[Asarin, 1995] Asarin, E. (1995). Chaos and undecidabilty (draft). Avalaible in *http ://www.liafa.jussieu.fr/˜asarin/*.

[Asarin, 1998] Asarin, E. (1998). Equations on timed languages. In Henzinger, T. A. and Sastry, S., editors, *Hybrid Systems : Computation and Control, First International Workshop, HSCC'98, Berkeley, California, USA, April 13-15, 1998, Proceedings*, volume 1386 of *Lecture Notes in Computer Science*, pages 1–12. Springer.

[Asarin, 2006] Asarin, E. (2006). Noise and decidability. Continuous Dynamics and Computability Colloquium. Video and sound available trough "Diffusion des savoirs de l'Ecole Normale Supérieure", on *http ://www.diffusion.ens.fr/en/index.php ?res=conf&idconf=1226*.

[Asarin and Bouajjani, 2001] Asarin, E. and Bouajjani, A. (2001). Perturbed Turing machines and hybrid systems. In *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science (LICS-01)*, pages 269–278, Los Alamitos, CA. IEEE Computer Society Press.

[Asarin et al., 1997] Asarin, E., Caspi, P., and Maler, O. (1997). A Kleene theorem for timed automata. In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science*, pages 160–171, Warsaw, Poland. IEEE Computer Society Press.

[Asarin et al., 2002] Asarin, E., Caspi, P., and Maler, O. (2002). Timed regular expressions. *Journal of the ACM*, 49(2) :172–206.

[Asarin and Dima, 2002] Asarin, E. and Dima, C. (2002). Balanced timed regular expressions. *Electronic Notes in Theoretical Computer Science*, 68(5).

[Asarin and Maler, 1998] Asarin, E. and Maler, O. (1998). Achilles and the tortoise climbing up the arithmetical hierarchy. *Journal of Computer and System Sciences*, 57(3) :389–398.

[Asarin et al., 1995] Asarin, E., Maler, O., and Pnueli, A. (1995). Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138(1) :35–65.

[Asarin and Schneider, 2002] Asarin, E. and Schneider, G. (2002). Widening the boundary between decidable and undecidable hybrid systems. In Brim, L., Jancar, P., Kretínský, M., and Kucera, A., editors, *CONCUR 2002 - Concurrency Theory, 13th International Conference, Brno, Czech Republic, August 20-23, 2002, Proceedings*, volume 2421 of *Lecture Notes in Computer Science*, pages 193–208. Springer.

[Asarin et al., 2001] Asarin, E., Schneider, G., and Yovine, S. (2001). On the decidability of the reachability problem for planar differential inclusions. In Benedetto, M. D. D. and Sangiovanni-Vincentelli, A. L., editors, *Hybrid Systems : Computation and Control, 4th International Workshop, HSCC 2001, Rome, Italy, March 28-30, 2001, Proceedings*, volume 2034 of *Lecture Notes in Computer Science*, pages 89–104. Springer.

[Beauquier, 1998] Beauquier, D. (1998). Pumping lemmas for timed automata. In Nivat, M., editor, *Foundations of Software Science and Computation Structure, First International Conference, FoSSaCS'98, Held as Part of the European Joint Conferences on the Theory and Practice of Software, ETAPS'98, Lisbon, Portugal, March 28 - April 4, 1998, Proceedings*, volume 1378 of *Lecture Notes in Computer Science*, pages 81–94. Springer.

[Ben-Hur et al., 2003] Ben-Hur, A., Feinberg, J., Fishman, S., and Siegelmann, H. T. (2003). Probabilistic analysis of a differential equation for linear programming. *Journal of Complexity*, 19(4) :474–510.

[Ben-Hur et al., 2004a] Ben-Hur, A., Feinberg, J., Fishman, S., and Siegelmann, H. T. (2004a). Random matrix theory for the analysis of the performance of an analog computer : a scaling theory. *Physics Letters A*, 323(3–4) :204–209.

[Ben-Hur et al., 2004b] Ben-Hur, A., Roitershtein, A., and Siegelmann, H. T. (2004b). On probabilistic analog automata. *Theoretical Computer Science*, 320(2–3) :449–464.

[Ben-Hur et al., 2002] Ben-Hur, A., Siegelmann, H. T., and Fishman, S. (2002). A theory of complexity for continuous time systems. *Journal of Complexity*, 18(1) :51–86.

[Blondel and Tsitsiklis, 1999] Blondel, V. D. and Tsitsiklis, J. N. (1999). Complexity of stability and controllability of elementary hybrid systems. *Automatica*, 35(3) :479–489.

[Blondel and Tsitsiklis, 2000] Blondel, V. D. and Tsitsiklis, J. N. (2000). A survey of computational complexity results in systems and control. *Automatica*, 36(9) :1249–1274.

[Blum et al., 1998] Blum, L., Cucker, F., Shub, M., and Smale, S. (1998). *Complexity and Real Computation.* Springer-Verlag.

[Blum et al., 1989] Blum, L., Shub, M., and Smale, S. (1989). On a theory of computation and complexity over the real numbers ; NP completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21(1) :1–46.

[Bournez, 1999a] Bournez, O. (1999a). Achilles and the Tortoise climbing up the hyper-arithmetical hierarchy. *Theoretical Computer Science*, 210(1) :21–71.

[Bournez, 1999b] Bournez, O. (1999b). *Complexité Algorithmique des Systèmes Dynamiques Continus et Hybrides*. PhD thesis, Ecole Normale Supérieure de Lyon.

[Bournez, 2006] Bournez, O. (2006). How much can analog and hybrid systems be proved (super-)Turing. *Applied Mathematics and Computation*, 178(1) :58–71.

[Bournez et al., 2006] Bournez, O., Campagnolo, M. L., Graça, D. S., and Hainry, E. (2006). The general purpose analog computer and computable analysis are two equivalent paradigms of analog computation. In Cai, J., Cooper, S. B., and Li, A., editors, *Theory and Applications of Models of Computation, Third International Conference, TAMC 2006, Beijing, China, May 15-20, 2006, Proceedings*, volume 3959 of *Lecture Notes in Computer Science*, pages 631–643. Springer.

[Bournez and Hainry, 2005] Bournez, O. and Hainry, E. (2005). Elementarily computable functions over the real numbers and $\mathbb{R}$-sub-recursive functions. *Theoretical Computer Science*, 348(2–3) :130–147.

[Bournez and Hainry, 2006] Bournez, O. and Hainry, E. (2006). Recursive analysis characterized as a class of real recursive functions. *Fundamenta Informaticae*, 74(4) :409–433.

[Bouyer et al., 2000a] Bouyer, P., Dufourd, C., Fleury, E., and Petit, A. (2000a). Are timed automata updatable ? In Emerson, E. A. and Sistla, A. P., editors, *Computer Aided Verification, 12th International Conference, CAV 2000, Chicago, IL, USA, July 15-19, 2000, Proceedings*, volume 1855 of *Lecture Notes in Computer Science*, pages 464–479. Springer.

[Bouyer et al., 2000b] Bouyer, P., Dufourd, C., Fleury, E., and Petit, A. (2000b). Expressiveness of updatable timed automata. In Nielsen, M. and Rovan, B., editors, *Mathematical Foundations of Computer Science 2000, 25th International Symposium, MFCS 2000, Bratislava, Slovakia, August 28 - September 1, 2000, Proceedings*, volume 1893 of *Lecture Notes in Computer Science*, pages 232–242. Springer.

[Bouyer and Petit, 1999] Bouyer, P. and Petit, A. (1999). Decomposition and composition of timed automata. In Wiedermann, J., van Emde Boas, P., and Nielsen, M., editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, volume 1644 of *Lecture Notes in Computer Science*, pages 210–219. Springer.

[Bouyer and Petit, 2002] Bouyer, P. and Petit, A. (2002). A Kleene/Büchi-like theorem for clock languages. *Journal of Automata, Languages and Combinatorics*, 7(2) :167–186.

[Bowles, 1996] Bowles, M. D. (1996). U.S. technological enthusiasm and British technological skepticism in the age of the analog brain. *IEEE Annals of the History of Computing*, 18(4) :5–15.

[Branicky, 1995a] Branicky, M. S. (1995a). *Studies in Hybrid Systems : Modeling, Analysis, and Control*. PhD thesis, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA.

[Branicky, 1995b] Branicky, M. S. (1995b). Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138(1) :67–100.

[Brihaye, 2006] Brihaye, T. (2006). A note on the undecidability of the reachability problem for o-minimal dynamical systems. *Math. Log. Q*, 52(2) :165–170.

[Brihaye and Michaux, 2005] Brihaye, Th. and Michaux, Ch. (2005). On the expressiveness and decidability of o-minimal hybrid systems. *Journal of Complexity*, 21(4) :447–478.

[Brockett, 1989] Brockett, R. W. (1989). Smooth dynamical systems which realize arithmetical and logical operations. In Nijmeijer, H. and Schumacher, J. M., editors, *Three Decades of Mathematical Systems Theory*, volume 135 of *Lecture Notes in Computer Science*, pages 19–30. Springer.

[Brockett, 1991] Brockett, R. W. (1991). Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems. *Linear Algebra and its Applications*, 146 :79–91.

[Brockett, 1994] Brockett, R. W. (1994). Dynamical systems and their associated automata. In U. Helmke, R. M. and Saurer, J., editors, *Systems and Networks : Mathematical Theory and Applications*, volume 77, pages 49–69. Akademi-Verlag, Berlin.

[Bush, 1931] Bush, V. (1931). The differential analyser. *Journal of the Franklin Institute*, 212(4) :447–488.

[Calude and Pavlov, 2002] Calude, C. S. and Pavlov, B. (2002). Coins, quantum measurements, and Turing's barrier. *Quantum Information Processing*, 1(1-2) :107–127.

[Campagnolo, 2002] Campagnolo, M. (2002). The complexity of real recursive functions. In Calude, C., Dinneen, M., and Peper, F., editors, *Unconventional Models of Computation, UMC'02*, number 2509 in Lecture Notes in Computer Science, pages 1–14. Springer-Verlag.

[Campagnolo, 2004] Campagnolo, M. (2004). Continuous time computation with restricted integration capabilities. *Theoretical Computer Science*, 317(4) :147–165.

[Campagnolo et al., 2000] Campagnolo, M., Moore, C., and Costa, J. F. (2000). Iteration, inequalities, and differentiability in analog computers. *Journal of Complexity*, 16(4) :642–660.

[Campagnolo et al., 2002] Campagnolo, M., Moore, C., and Costa, J. F. (2002). An analog characterization of the Grzegorczyk hierarchy. *Journal of Complexity*, 18(4) :977–1000.

[Campagnolo, 2001] Campagnolo, M. L. (2001). *Computational complexity of real valued recursive functions and analog circuits*. PhD thesis, IST, Universidade Técnica de Lisboa.

[Campagnolo and Ojakian, 2007] Campagnolo, M. L. and Ojakian, K. (2007). The methods of approximation and lifting in real computation. In Cenzer, D., Dillhage, R., Grubba, T., and Weihrauch, K., editors, *Proceedings of the Third International Conference on Computability and Complexity in Analysis, CCA 2006, Gainesville, Florida, USA, November 1–5, 2006*, volume 167 of *Electronic Notes in Theoretical Computer Science*, Amsterdam. Elsevier.

[Casey, 1996] Casey, M. (1996). The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction. *Neural Computation*, 8 :1135–1178.

[Casey, 1998] Casey, M. (1998). Correction to proof that recurrent neural networks can robustly recognize only regular languages. *Neural Computation*, 10 :1067–1069.

[Ceraens and Viksna, 1996] Ceraens, K. and Viksna, J. (1996). Deciding reachability for planar multi-polynomial systems. In *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, page 389. Springer-Verlag.

[Church, 1936] Church, A. (1936). An unsolvable problem of elementary number theory. *American Journal of Mathematics,*, 58 :345–363. Reprinted in [Davis, 1965].

[Clote, 1998] Clote, P. (1998). Computational models and function algebras. In Griffor, E. R., editor, *Handbook of Computability Theory*, pages 589–681. North-Holland, Amsterdam.

[Coddington and Levinson, 1972] Coddington, E. A. and Levinson, N. (1972). *Theory of Ordinary Differentiel Equations*. McGraw-Hill.

[Cohen and Grossberg, 1983] Cohen, M. A. and Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(5) :815–826.

[Collins, 2005] Collins, P. (2005). Continuity and computability on reachable sets. *Theoretical Computer Science*, 341 :162–195.

[Collins and Lygeros, 2005] Collins, P. and Lygeros, J. (2005). Computability of finite-time reachable sets for hybrid systems. In *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference*, pages 4688–4693. IEEE Computer Society Press.

[Collins and van Schuppen, 2004] Collins, P. and van Schuppen, J. H. (2004). Observability of piecewise-affine hybrid systems. In Alur, R. and Pappas, G. J., editors, *Hybrid Systems : Computation and Control, 7th International Workshop, HSCC 2004, Philadelphia, PA, USA, March 25-27, 2004, Proceedings*, volume 2993 of *Lecture Notes in Computer Science*, pages 265–279. Springer.

[Copeland, 1998] Copeland, B. J. (1998). Even Turing machines can compute uncomputable functions. In Calude, C., Casti, J., and Dinneen, M., editors, *Unconventional Models of Computations*. Springer-Verlag.

[Copeland, 2002] Copeland, B. J. (2002). Accelerating Turing machines. *Minds and Machines*, 12 :281–301.

[Costa and Mycka, 2006] Costa, J. F. and Mycka, J. (2006). The conjecture $P \neq NP$ given by some analytic condition. In Bekmann, A., Berger, U., Löwe, B., and Tucker, J., editors, *Logical Approaches to Computational Barriers, Second conference on Computability in Europe, CiE 2006*, pages 47–57, Swansea, UK. Report CSR 7-26, Report Series, University of Wales Swansea Press, 2006.

[Coward, 2006] Coward, D. (2006). Doug coward's analog computer museum. *http ://dco-ward.best.vwh.net/analog/.*

[Davies, 2001] Davies, E. B. (2001). Building infinite machines. *The British Journal for the Philosophy of Science*, 52 :671–682.

[Davis, 1965] Davis, M. (1965). *The Undecidable : Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions.* Raven Press. Reprinted by Dover Publications, Incorporated in 2004.

[Dee and Ghil, 1984] Dee, D. and Ghil, M. (1984). Boolean difference equations, I : Formulation and dynamic behavior. *SIAM Journal on Applied Mathematics*, 44(1) :111–126.

[Delvenne et al., 2004] Delvenne, J.-C., Kurka, P., and Blondel, V. D. (2004). Computational universality in symbolic dynamical systems. In Margenstern, M., editor, *MCU : International Conference on Machines, Computations, and Universality*, volume 3354 of *Lecture Notes in Computer Science*, pages 104–115. Springer.

[Deutsch, 1985] Deutsch, D. (1985). Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society (London), Serie A*, 400 :97–117.

[Durand-Lose, 2005] Durand-Lose, J. (2005). Abstract geometrical computation : Turing-computing ability and undecidability. In Cooper, S. B., Löwe, B., and Torenvliet, L., editors, *New Computational Paradigms, First Conference on Computability in Europe, CiE 2005, Amsterdam, The Netherlands, June 8-12, 2005, Proceedings*, volume 3526 of *Lecture Notes in Computer Science*, pages 106–116. Springer.

[Earman and Norton, 1993] Earman, J. and Norton, J. D. (1993). Forever is a day : Supertasks in Pitowksy and Malament-Hogarth spacetimes. *Philosophy of Science*, 60(1) :22–42.

[Etesi and Németi, 2002] Etesi, G. and Németi, I. (2002). Non-Turing computations via Malament-Hogarth space-times. *International Journal Theoretical Physics*, 41 :341–370.

[Faybusovich, 1991a] Faybusovich, L. (1991a). Dynamical systems which solve optimization problems with linear constraints. *IMA Journal of Mathematical Control and Information*, 8 :135–149.

[Faybusovich, 1991b] Faybusovich, L. (1991b). Hamiltonian structure of dynamical systems which solve linear programming problems. *Physics*, D53 :217–232.

[Filippov, 1988] Filippov, A. (1988). *Differential equations with discontinuous right-hand sides.* Kluwer Academic Publishers.

[Finkel, 2005] Finkel, O. (2005). On decision problems for timed automata. *Bulletin of the EATCS*, 87 :185–190.

[Finkel, 2006] Finkel, O. (2006). On the shuffle of regular timed languages. *Bulletin of the European Association for Theoretical Computer Science*, 88 :182–184. Technical Contributions.

[Foy, 2004] Foy, J. (2004). A dynamical system which must be stable whose stability cannot be proved. *Theoretical Computer Science*, 328(3) :355–361.

[Francisco, 2002] Francisco, A. P. L. (2002). Finite automata over continuous time. Diploma Thesis. Universidade Técnica de Lisboa, Instituto Superior Técnico.

[Fränzle, 1999] Fränzle, M. (1999). Analysis of hybrid systems : An ounce of realism can save an infinity of states. In Flum, J. and Rodríguez-Artalejo, M., editors, *Computer Science Logic (CSL'99)*, volume 1683 of *Lecture Notes in Computer Science*, pages 126–140. Springer Verlag.

[Gori and Meer, 2002] Gori, M. and Meer, K. (2002). A step towards a complexity theory for analog systems. *Mathematical Logic Quarterly*, 48(Suppl. 1) :45–58.

[Graça et al., 2005] Graça, D., Campagnolo, M., and Buescu, J. (2005). Robust simulations of Turing machines with analytic maps and flows. In Cooper, B., Loewe, B., and Torenvliet, L., editors, *Proceedings of CiE'05, New Computational Paradigms*, volume 3526 of *Lecture Notes in Computer Science*, pages 169–179. Springer-Verlag.

[Graça et al., 2007] Graça, D. S., Campagnolo, M. L., and Buescu, J. (2007). Computability with polynomial differential equations. *Advances in Applied Mathematics.* To appear.

[Graça et al., 2006] Graça, D. S., Zhong, N., and Buescu, J. (2006). Computability, noncomputability and undecidability of maximal intervals of IVPs. *Transactions of the American Mathematical Society*. To appear.

[Graça, 2002] Graça, D. (2002). The general purpose analog computer and recursive functions over the reals. Master's thesis, IST, Universidade Técnica de Lisboa.

[Graça, 2004] Graça, D. S. (2004). Some recent developments on Shannon's general purpose analog computer. *Mathematical Logic Quarterly*, 50(4–5) :473–485.

[Graça and Costa, 2003] Graça, D. S. and Costa, J. F. (2003). Analog computers and recursive functions over the reals. *Journal of Complexity*, 19(5) :644–664.

[Grigorieff and Margenstern, 2004] Grigorieff, S. and Margenstern, M. (2004). Register cellular automata in the hyperbolic plane. *Fundamenta Informaticae*, 1(61) :19–27.

[Gruska, 1997] Gruska, J. (1997). *Foundations of Computing*. International Thomson Publishing.

[Gupta et al., 1997] Gupta, V., Henzinger, T. A., and Jagadeesan, R. (1997). Robust timed automata. In Maler, O. O., editor, *Hybrid and real-time systems : international workshop, HART '97, Grenoble, France, March 26–28, 1997*, volume 1201 of *Lecture Notes in Computer Science*, pages 331–345, Berlin. Springer-Verlag.

[Head, 1987] Head, T. (1987). Formal language theory and DNA : An analysis of the generative capacity of specific recombinant behaviors. *Bulletin of Mathematical Biology*, 49 :737–759.

[Helmke and Moore, 1994] Helmke, U. and Moore, J. (1994). *Optimization and Dynamical Systems*. Communications and Control Engineering Series. Springer Verlag, London.

[Henzinger and Raskin, 1999] Henzinger, T. and Raskin, J.-F. (1999). Robust undecidability of timed and hybrid systems. In Vaandrager, F. and van Schuppen, J. H., editors, *Hybrid systems : computation and control ; Second International Workshop, HSCC'99, Berg en Dal, The Netherlands, March 29–31, 1999 ; proceedings*, volume 1569 of *Lecture Notes in Computer Science*, Berlin. Springer-Verlag.

[Henzinger et al., 1998] Henzinger, T. A., Kopke, P. W., Puri, A., and Varaiya, P. (1998). What's decidable about hybrid automata ? *Journal of Computer and System Sciences*, 57(1) :94–124.

[Hirsch et al., 2003] Hirsch, M. W., Smale, S., and Devaney, R. (2003). *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Elsevier Academic Press.

[Hogarth, 1994] Hogarth, M. (1994). Non-Turing computers and non-Turing computability. In *Proceedings of the Philosophy of Science Association (PSA'94)*, volume 1, pages 126–138.

[Hogarth, 1996] Hogarth, M. (1996). *Predictability, Computability and Spacetime*. PhD thesis, Sidney Sussex College, Cambridge.

[Hogarth, 2006] Hogarth, M. (2006). Non-Turing computers are the new non-Eucliedean geometries. In *Future Trends in Hypercomputation*. Sheffield, 11–13 September 2006. Available for download on www.hypercomputation.net.

[Hogarth, 1992] Hogarth, M. L. (1992). Does general relativity allow an observer to view an eternity in a finite time ? *Foundations of Physics Letters*, 5 :173–181.

[Hopfield, 1984] Hopfield, J. J. (1984). Neural networks with graded responses have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 81 :3088–3092.

[Hopfield and Tank, 1985] Hopfield, J. J. and Tank, D. W. (1985). 'Neural' computation of decisions in optimization problems. *Biological Cybernetics*, 52 :141–152.

[Hoyrup, 2006] Hoyrup, M. (2006). Dynamical systems : stability and simulability. Technical report, Département d'Informatique, ENS Paris.

[Kempe, 1876] Kempe, A. (1876). On a general method of describing plane curves of the n–th degree by linkwork. *Proceedings of the London Mathematical Society*, 7 :213–216.

[Kieu, 2004] Kieu, T. D. (2004). Hypercomputation with quantum adiabatic processes. *Theoretical Computer Science*, 317(1-3) :93–104.

[Kleene, 1936] Kleene, S. C. (1936). General recursive functions of natural numbers. *Mathematical Annals*, 112 :727–742. Also in [Davis, 1965].

[Ko, 1983] Ko, K.-I. (1983). On the computational complexity of ordinary differential equations. *Information and Control*, 58(1-3) :157–194.

[Ko, 1991] Ko, K.-I. (1991). *Complexity Theory of Real Functions*. Progress in Theoretical Computer Science. Birkhäuser, Boston.

[Koiran, 2001] Koiran, P. (2001). The topological entropy of iterated piecewise affine maps is uncomputable. *Discrete Mathematics & Theoretical Computer Science*, 4(2) :351–356.

[Koiran et al., 1994] Koiran, P., Cosnard, M., and Garzon, M. (1994). Computability with low-dimensional dynamical systems. *Theoretical Computer Science*, 132(1-2) :113–128.

[Koiran and Moore, 1999] Koiran, P. and Moore, C. (1999). Closed-form analytic maps in one and two dimensions can simulate universal Turing machines. *Theoretical Computer Science*, 210(1) :217–223.

[Korovina and Vorobjov, 2004] Korovina, M. V. and Vorobjov, N. (2004). Pfaffian hybrid systems. In Marcinkowski, J. and Tarlecki, A., editors, *Computer Science Logic, 18th International Workshop, CSL 2004, 13th Annual Conference of the EACSL, Karpacz, Poland, September 20-24, 2004, Proceedings*, volume 3210 of *Lecture Notes in Computer Science*, pages 430–441. Springer.

[Korovina and Vorobjov, 2006] Korovina, M. V. and Vorobjov, N. (2006). Upper and lower bounds on sizes of finite bisimulations of Pfaffian hybrid systems. In Beckmann, A., Berger, U., Löwe, B., and Tucker, J. V., editors, *Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, CiE 2006, Swansea, UK, June 30-July 5, 2006, Proceedings*, volume 3988 of *Lecture Notes in Computer Science*, pages 267–276. Springer.

[Kurganskyy and Potapov, 2005] Kurganskyy, O. and Potapov, I. (2005). Computation in one-dimensional piecewise maps and planar pseudo-billiard systems. In Calude, C., Dinneen, M. J., Paun, G., Pérez-Jiménez, M. J., and Rozenberg, G., editors, *Unconventional Computation, 4th International Conference, UC 2005, Sevilla, Spain, October 3-7, 2005, Proceedings*, volume 3699 of *Lecture Notes in Computer Science*, pages 169–175. Springer.

[Lafferriere and Pappas, 2000] Lafferriere, G. and Pappas, G. J. (2000). O-minimal hybrid systems. *Mathematics of Control, Signals, and Systems*, 13 :1–21.

[Legenstein and Maass, 2007] Legenstein, R. and Maass, W. (2007). What makes a dynamical system computationally powerful ? In Haykin, S., Principe, J. C., Sejnowski, T., and McWhirter, J., editors, *New Directions in Statistical Signal Processing : From Systems to Brain*, pages 127–154. MIT Press.

[Lipshitz and Rubel, 1987] Lipshitz, L. and Rubel, L. A. (1987). A differentially algebraic replacement theorem, and analog computability. *Proceedings of the American Mathematical Society*, 99(2) :367–372.

[Lipton, 1995] Lipton, R. J. (1995). DNA solution of hard computational problems. *Science*, 268 :542–545.

[Loff, 2007] Loff, B. (2007). A functional characterisation of the analytical hierarchy. In *Computability in Europe 2007 : Computation and Logic in the Real World.*

[Loff et al., 2007a] Loff, B., Costa, J. F., and Mycka, J. (2007a). Computability on reals, infinite limits and differential equations. *Applied Mathematics and Computation*. To appear.

[Loff et al., 2007b] Loff, B., Costa, J. F., and Mycka, J. (2007b). The new promise of analog computation. In *Computability in Europe 2007 : Computation and Logic in the Real World.*

[Maass, 1996a] Maass, W. (1996a). Lower bounds for the computational power of networks of spiking neurons. *Neural Computation*, 8(1) :1–40.

[Maass, 1996b] Maass, W. (1996b). On the computational power of noisy spiking neurons. In Touretzky, D., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems*, volume 8, pages 211–217. MIT Press (Cambridge).

[Maass, 1997a] Maass, W. (1997a). A model for fast analog computations with noisy spiking neurons. In Bower, J., editor, *Computational Neuroscience : Trends in research*, pages 123–127.

[Maass, 1997b] Maass, W. (1997b). Networks of spiking neurons : the third generation of neural network models. *Neural Networks*, 10 :1659–1671.

[Maass, 1999] Maass, W. (1999). Computing with spiking neurons. In Maass, W. and Bishop, C. M., editors, *Pulsed Neural Networks*, pages 55–85. MIT Press (Cambridge).

[Maass, 2002] Maass, W. (2002). Computing with spikes. *Special Issue on Foundations of Information Processing of TELEMATIK*, 8(1) :32–36.

[Maass, 2003] Maass, W. (2003). Computation with spiking neurons. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*, pages 1080–1083. MIT Press (Cambridge), 2nd edition.

[Maass and Bishop, 1998] Maass, W. and Bishop, C. (1998). *Pulsed Neural Networks*. Cambridge MA. MIT Press.

[Maass et al., 2007] Maass, W., Joshi, P., and Sontag, E. D. (2007). Computational aspects of feedback in neural circuits. *Public Library of Science Computational Biology*, 3(1). e165.

[Maass and Natschläger, 2000] Maass, W. and Natschläger, T. (2000). A model for fast analog computation based on unreliable synapses. *Neural Computation*, 12(7) :1679–1704.

[Maass and Orponen, 1998] Maass, W. and Orponen, P. (1998). On the effect of analog noise in discrete-time analog computations. *Neural Computation*, 10(5) :1071–1095.

[Maass and Ruf, 1999] Maass, W. and Ruf, B. (1999). On computation with pulses. *Information and Computation*, 148(2) :202–218.

[Maass and Sontag, 1999] Maass, W. and Sontag, E. (1999). Analog neural nets with gaussian or other common noise distributions cannot recognize arbitrary regular languages. *Neural Computation*, 11(3) :771–782.

[MacLennan, 2001] MacLennan, B. J. (2001). Can differential equations compute. Technical report, Computer Science Department, University of Tennesse, Knoxville.

[Mills, 1995] Mills, J. (1995). Programmable VLSI extended analog computer for cyclotron beam control. Technical Report 441, Indiana University Computer Science.

[Mills et al., 2005] Mills, J. W., Himebaugh, B., Allred, A., Bulwinkle, D., Deckard, N., Gopalakrishnan, N., Miller, J., Miller, T., Nagai, K., Nakamura, J., Ololoweye, B., Vlas, R., Whitener, P., Ye, M., , and Zhang, C. (2005). Extended analog computers : A unifying paradigm for VLSI, plastic and colloidal computing systems. In *Workshop on Unique Chips and Systems (UCAS-1). Held in conjunction with IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS05)*, Austin, Texas.

[Minsky, 1967] Minsky, M. L. (1967). *Computation : Finite and infinite machines*. Prentice-Hall, Englewood Cliffs.

[Moore, 1990] Moore, C. (1990). Unpredictability and undecidability in dynamical systems. *Physical Review Letters*, 64(20) :2354–2357.

[Moore, 1991] Moore, C. (1991). Generalized shifts : unpredictability and undecidability in dynamical systems. *Nonlinearity*, 4(3) :199–230.

[Moore, 1996] Moore, C. (1996). Recursion theory on the reals and continuous-time computation. *Theoretical Computer Science*, 162(1) :23–44.

[Moore, 1998a] Moore, C. (1998a). Dynamical recognizers : real-time language recognition by analog computers. *Theoretical Computer Science*, 201(1–2) :99–136.

[Moore, 1998b] Moore, C. (1998b). Finite-dimensional analog computers : Flows, maps, and recurrent neural networks. In Calude, C. S., Casti, J. L., and Dinneen, M. J., editors, *Unconventional Models of Computation (UMC'98)*. Springer.

[Murray, 2002] Murray, J. D. (2002). Mathematical biology. I : An introduction. In *Biomathematics*, volume 17. Springer Verlag, third edition.

[Mycka and Costa, 2004] Mycka, J. and Costa, J. F. (2004). Real recursive functions and their hierarchy. *Journal of Complexity*, 20(6) :835–857.

[Mycka and Costa, 2005] Mycka, J. and Costa, J. F. (2005). What lies beyond the mountains ? computational systems beyond the Turing limit. *European Association for Theoretical Computer Science Bulletin*, 85 :181–189.

[Mycka and Costa, 2006] Mycka, J. and Costa, J. F. (2006). The $P \neq NP$ conjecture in the context of real and complex analysis. *Journal of Complexity*, 22(2) :287–303.

[Mycka and Costa, 2007] Mycka, J. and Costa, J. F. (2007). A new conceptual framework for analog computation. *Theoretical Computer Science*, 374 :277–290.

[Müller and Moiske, 1993] Müller, N. and Moiske, B. (1993). Solving initial value problems in polynomial time. In *Proc. 22 JAIIO - PANEL '93, Part 2*, pages 283–293.

[Natschläger and Maass, 2002] Natschläger, T. and Maass, W. (2002). Spiking neurons and the induction of finite state machines. *Theoretical Computer Science : Special Issue on Natural Computing*, 287(1) :251–265.

[Németi and Andréka, 2006] Németi, I. and Andréka, H. (2006). New physics and hypercomputation. In Wiedermann, J., Tel, G., Pokorný, J., Bieliková, M., and Stuller, J., editors, *SOFSEM 2006 : Theory and Practice of Computer Science, 32nd Conference on Current Trends in Theory and Practice of Computer Science, Merín, Czech Republic, January 21-27, 2006, Proceedings*, volume 3831 of *Lecture Notes in Computer Science*, page 63. Springer.

[Németi and Dávid, 2006] Németi, I. and Dávid, G. (2006). Relativistic computers and the Turing barrier. *Applied Mathematics and Computation*, 178 :118–142.

[Nicollin et al., 1993] Nicollin, X., Olivero, A., Sifakis, J., and Yovine, S. (1993). An approach to the description and analysis of hybrid systems. In Grossman, R. L., Nerode, A., Ravn, A. P., and Rischel, H., editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 149–178. Springer-Verlag.

[Omohundro, 1984] Omohundro, S. (1984). Modelling cellular automata with partial differential equations. *Physica D*, 10D(1–2) :128–134.

[Orponen, 1994] Orponen, P. (1994). Computational complexity of neural networks : a survey. *Nordic Journal of Computing*, 1(1) :94–110.

[Orponen, 1996] Orponen, P. (1996). The computational power of discrete Hopfield nets with hidden units. *Neural Computation*, 8(2) :403–415.

[Orponen, 1997] Orponen, P. (1997). A survey of continous-time computation theory. In *Advances in Algorithms, Languages, and Complexity*, pages 209–224. Kluwer Academic Publishers.

[Orponen and Šíma, 2000] Orponen, P. and Šíma, J. (2000). A continuous-time Hopfield net simulation of discrete neural networks. In *Proceedings of the 2nd International ICSC Symposium on Neural Computations (NC'2000)*, pages 36–42, Berlin, Germany. Wetaskiwin (Canada) : ICSC Academic Press.

[Papadimitriou, 2001] Papadimitriou, C. (2001). Algorithms, games, and the Internet. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing : Hersonissos, Crete, Greece, July 6–8, 2001*, pages 749–753, New York, NY, USA. ACM Press.

[Păun, 2002] Păun, G. (2002). *Membrane Computing. An Introduction.* Springer-Verlag, Berlin.

[Post, 1946] Post, E. (1946). A variant of a recursively unsolvable problem. *Bulletin of the American Math. Soc.*, 52 :264–268.

[Pour-El and Zhong, 1997] Pour-El, M. and Zhong, N. (1997). The wave equation with computable initial data whose unique solution is nowhere computable. *Mathematical Logic Quarterly*, 43(4) :499–509.

[Pour-El, 1974] Pour-El, M. B. (1974). Abstract computability and its relation to the general purpose analog computer (some connections between logic, differential equations and analog computers). *Transactions of the American Mathematical Society*, 199 :1–28.

[Pour-El and Richards, 1979] Pour-El, M. B. and Richards, J. I. (1979). A computable ordinary differential equation which possesses no computable solution. *Annals of Mathematical Logic*, 17 :61–90.

[Pour-El and Richards, 1981] Pour-El, M. B. and Richards, J. I. (1981). The wave equation with computable initial data such that its unique solution is not computable. *Advances in Mathematics*, 39 :215–239.

[Pour-El and Richards, 1989] Pour-El, M. B. and Richards, J. I. (1989). *Computability in Analysis and Physics*. Springer-Verlag.

[Puri, 1998] Puri, A. (1998). Dynamical properties of timed automata. In Ravn, A. P. and Rischel, H., editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems, 5th International Symposium, FTRTFT'98, Lyngby, Denmark, September 14-18, 1998, Proceedings*, volume 1486 of *Lecture Notes in Computer Science*, pages 210–227. Springer.

[Puri and Varaiya, 1994] Puri, A. and Varaiya, P. (1994). Decidability of hybrid systems with rectangular differential inclusions. In *Proceedings of the 6th workshop on Computer-aided verification*, volume 818 of *Lecture Notes in Computer Science*, pages 95–104. Springer-Verlag.

[Rabin, 1963] Rabin, M. O. (1963). Probabilistic automata. *Information and Control*, 6(3) :230–245.

[Rabinovich, 2003] Rabinovich, A. (2003). Automata over continuous time. *Theoretical Computer Science*, 300(1–3) :331–363.

[Rabinovich and Trakhtenbrot, 1997] Rabinovich, A. M. and Trakhtenbrot, B. A. (1997). From finite automata toward hybrid systems (extended abstract). In Chlebus, B. S. and Czaja, L., editors, *FCT*, volume 1279 of *Lecture Notes in Computer Science*, pages 411–422. Springer.

[Rubel, 1989] Rubel, L. A. (1989). A survey of transcendentally transcendental functions. *American Mathematical Monthly*, 96(9) :777–788.

[Rubel, 1993] Rubel, L. A. (1993). The extended analog computer. *Advances in Applied Mathematics*, 14 :39–50.

[Ruohonen, 1993] Ruohonen, K. (1993). Undecidability of event detection for ODEs. *Journal of Information Processing and Cybernetics*, 29 :101–113.

[Ruohonen, 1994] Ruohonen, K. (1994). Event detection for ODEs and nonrecursive hierarchies. In Karhumäki, J. and Maurer, H., editors, *Proceedings of the Colloquium in Honor of Arto Salomaa. Results and Trends in Theoretical Computer Science (Graz, Austria, June 10-11, 1994)*, volume 812 of *Lecture Notes in Computer Science*, pages 358–371. Springer-Verlag, Berlin.

[Ruohonen, 1996] Ruohonen, K. (1996). An effective Cauchy-Peano existence theorem for unique solutions. *International Journal of Foundations of Computer Science*, 7(2) :151–160.

[Ruohonen, 1997a] Ruohonen, K. (1997a). Decidability and complexity of event detection problems for ODEs. *Complexity*, 2(6) :41–53.

[Ruohonen, 1997b] Ruohonen, K. (1997b). Undecidable event detection problems for ODEs of dimension one and two. *Theoretical Informatics and Applications*, 31(1) :67–79.

[Ruohonen, 2004] Ruohonen, K. (2004). Chomskian hierarchies of families of sets of piecewise continuous functions. *Theory of Computing Systems*, 37(5) :609–638.

[Shannon, 1941] Shannon, C. E. (1941). Mathematical theory of the differential analyser. *Journal of Mathematics and Physics MIT*, 20 :337–354.

[Shor, 1994] Shor, P. W. (1994). Algorithms for quantum computation : Discrete logarithms and factoring. In Goldwasser, S., editor, *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Los Alamitos, CA, USA. IEEE Computer Society Press.

[Siegelmann and Fishman, 1998] Siegelmann, H. T. and Fishman, S. (1998). Analog computation with dynamical systems. *Physica D*, 120 :214–235.

[Siegelmann and Sontag, 1994] Siegelmann, H. T. and Sontag, E. D. (1994). Analog computation via neural networks. *Theoretical Computer Science*, 131(2) :331–360.

[Siegelmann and Sontag, 1995] Siegelmann, H. T. and Sontag, E. D. (1995). On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1) :132–150.

[Šíma and Orponen, 2003a] Šíma and Orponen (2003a). Exponential transients in continuous-time Liapunov systems. *Theoretical Computer Science*, 306(1–3) :353–372.

[Šíma and Orponen, 2003b] Šíma, J. and Orponen, P. (2003b). Continuous-time symmetric Hopfield nets are computationally universal. *Neural Computation*, 15(3) :693–733.

[Šíma and Orponen, 2003c] Šíma, J. and Orponen, P. (2003c). General-purpose computation with neural networks : A survey of complexity theoretic results. *Neural Computation*, 15(12) :2727–2778.

[Smith, 2006] Smith, W. D. (2006). Church's thesis meets the N-body problem. *Applied Mathematics and Computation*, 178(1) :154–183.

[Stoll and Lee, 1988] Stoll, H. M. and Lee, L. S. (1988). A continuous-time optical neural network. In *IEEE Second International Conference on Neural Networks (2nd ICNN'88)*, volume II, pages 373–384, San Diego, CA. IEEE Society Press.

[Svoboda, 1948] Svoboda, A. (1948). *Computing Mechanisms and Linkages*. McGraw Hill. Dover reprint 1965.

[Thomson, 1876] Thomson, W. (1876). On an instrument for calculating the integral of the product of two given functions. In *Proceedings of the Royal Society of London*, volume 24, pages 266–276.

[Trakhtenbrot, 1995] Trakhtenbrot, B. (1995). Origins and metamorphoses of the trinity : Logic, nets, automata. In Kozen, D., editor, *Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science (San Diego, California, June 26-29, 1995)*, pages 506–507, Los Alamitos. IEEE Computer Society, IEEE Computer Society Press.

[Trakhtenbrot, 1999] Trakhtenbrot, B. A. (1999). Automata and their interaction : Definitional suggestions. In Ciobanu, G. and Paun, G., editors, *FCT*, volume 1684 of *Lecture Notes in Computer Science*, pages 54–89. Springer.

[Tripakis, 2003] Tripakis, S. (2003). Folk theorems on the determinization and minimization of timed automata. In Larsen, K. G. and Niebert, P., editors, *Formal Modeling and Analysis of Timed Systems : First International Workshop, FORMATS 2003, Marseille, France, September 6-7, 2003. Revised Papers*, volume 2791 of *Lecture Notes in Computer Science*, pages 182–188. Springer.

[Tucker and Zucker, 2007] Tucker, J. V. and Zucker, J. I. (2007). Computability of analog networks. *Theoretical Computer Science*, 371(1-2) :115–146.

[Turing, 1936] Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2) :230–265. Reprinted in [Davis, 1965].

[Vergis et al., 1986] Vergis, A., Steiglitz, K., and Dickinson, B. (1986). The complexity of analog computation. *Mathematics and Computers in Simulation*, 28(2) :91–113.

[Weihrauch, 2000] Weihrauch, K. (2000). *Computable Analysis*. Springer.

[Weihrauch and Zhong, 2002] Weihrauch, K. and Zhong, N. (2002). Is wave propagation computable or can wave computers beat the Turing machine ? *Proceedings of the London Mathematical Society*, 85(3) :312–332.

[Welch, 2006] Welch, P. D. (2006). The extent of computation in Malament-Hogarth spacetimes. http ://www.citebase.org/abstract ?id=oai :arXiv.org :gr-qc/0609035.

[Williams, 1996] Williams, M. R. (1996). About this issue. *IEEE Annals of the History of Computing*, 18(4).

[Woods and Naughton, 2005] Woods, D. and Naughton, T. J. (2005). An optical model of computation. *Theoretical Computer Science*, 334(1-3) :227–258.